

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2001年8月2日 (02.08.2001)

PCT

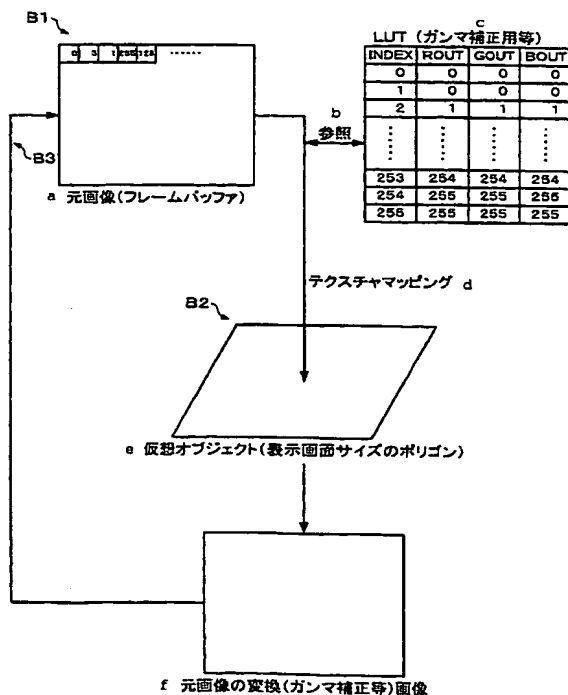
(10) 国際公開番号
WO 01/55969 A1

- (51) 国際特許分類⁷: G06T 15/00, A63F 13/00, G09G 5/00 (71) 出願人 (米国を除く全ての指定国について): 株式会社 ナムコ (NAMCO LTD.) [JP/JP]; 〒146-0095 東京都大田区多摩川2丁目8番5号 Tokyo (JP).
- (21) 国際出願番号: PCT/JP01/00408
- (22) 国際出願日: 2001年1月23日 (23.01.2001) (72) 発明者; および
- (25) 国際出願の言語: 日本語 (75) 発明者/出願人 (米国についてのみ): 橘高 繁 (KIT-SUTAKA, Shigeru) [JP/JP]; 〒146-0095 東京都大田区多摩川2丁目8番5号 株式会社 ナムコ内 Tokyo (JP).
- (26) 国際公開の言語: 日本語 (74) 代理人: 布施行夫, 外(FUSE, Yukio et al.); 〒167-0051 東京都杉並区荻窪5丁目26番13号 荻窪TMビル2階 Tokyo (JP).
- (30) 優先権データ: (81) 指定国 (国内): GB, US.
- 特願2000-20464 2000年1月28日 (28.01.2000) JP
特願2000-213725 2000年7月14日 (14.07.2000) JP
特願2000-213988 2000年7月14日 (14.07.2000) JP

[続葉有]

(54) Title: GAME SYSTEM AND IMAGE CREATING METHOD

(54) 発明の名称: ゲームシステム及び画像生成方法



(57) Abstract: A game system and a program for creating a focused image such as an image corrected by a video filter, for example, γ -correction or an image viewed in the real world with a light processing load. Original image information (R, G, B, and Z values) is defined as index numbers for a look-up table (LUT) for index color texture mapping. The look-up table (LUT) is used for texture mapping of a polygon having a display screen size (division block size) to conduct γ -correction. The Z value of each pixel of the original picture is defined as an index number of the look-up table (LUT). Texture mapping of a virtual object is conducted. The α value is so determined as to correspond to the Z value of each pixel of the original image. The original image and a blurred image are combined. Adjustment data for adjusting the luminance of the monitor is determined according to the operation data inputted by the player by means of a game controller and saved in a saving information storage device to convert the original image according to the adjustment data.

[続葉有]

WO 01/55969 A1



添付公開書類:
— 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

ガンマ補正などのビデオフィルタや現実世界の視界画像のようにフォーカシングされた画像を少ない処理負担で生成できるゲームシステム、プログラム等を提供する。元画像情報（R、G、B、Z値）をインデックスカラー・テクスチャマッピング用ルックアップテーブルLUTのインデックス番号として設定し、LUTを用いて表示画面サイズ（分割ブロックサイズ）のポリゴンにテクスチャマッピングを行い、ガンマ補正等を実現する。元画像の各画素のZ値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルLUTのインデックス番号に設定して仮想オブジェクトに対するテクスチャマッピングを行い、元画像の各画素のZ値に応じた値に α 値を設定し、元画像とそのぼかし画像を合成する。プレーヤがゲームコントローラを用いて入力した操作データに基づきモニタの明るさの調整データを設定し、設定された調整データをセーブ用情報記憶装置にセーブし、調整データに基づき元画像を変換する。

明 細 書

ゲームシステム及び画像生成方法

5 [技術分野]

本発明は、ゲームシステム、プログラム及び画像生成方法に関する。

[背景技術]

従来より、仮想的な 3 次元空間であるオブジェクト空間内の所与の視点から
10 見える画像を生成するゲームシステムが知られており、いわゆる仮想現実を体験できるものとして人気が高い。レーシングゲームを楽しむことができるゲームシステムを例にとれば、プレーヤは、車（オブジェクト）を操作してオブジェクト空間内で走行させ、他のプレーヤやコンピュータが操作する車と競争することで 3 次元ゲームを楽しむ。

15 さて、このようなゲームシステムでは、モニタ（表示部）の非線形特性を補正するために、画像に対してガンマ補正と呼ばれる変換を施すことが望ましい。

そして、このようなガンマ補正を実現する手法としては、以下に説明する第 1、第 2 の手法がある。

第 1 の手法では、図 1 A に示すように、メインメモリ 8 0 2 上にガンマ補正
20 用のルックアップテーブル（LUT）を用意しておく。そして、CPU 8 0 0（CPU 上で動作するソフトウェア）は、VRAM 8 0 6 内にあるフレームバッファ 8 0 8 から、元画像の各画素の色情報（RGB）を読み出す。そして、読み出された色情報に基づいてガンマ補正用 LUT を参照し、ガンマ補正後の色情報を得る。次に、得られたガンマ補正後の色情報をフレームバッファの対応する画素に書き戻す。そして、以上の処理を、元画像の全ての画素に対して
25 行う。

一方、第 2 の手法では、図 1 B に示すように、CPU 8 1 0 の制御下で動作

する描画プロセッサ 8 1 2 の後段に、ハードウェアによりガンマ補正を実現するガンマ補正回路 8 1 4 を設ける。そして、描画プロセッサ 8 1 2 により生成された色情報に対して、ガンマ補正回路 8 1 4 がガンマ補正を施し、モニタ 8 1 6 に出力する。

- 5 しかしながら、図 1 A の第 1 の手法では、フレームバッファ 8 0 8 からの色情報の読み出し、ガンマ補正用 L U T 8 0 4 の参照、ガンマ補正用 L U T 8 0 4 からの色情報の読み出し、読み出した色情報のフレームバッファ 8 0 8 への書き戻しなどの全ての処理を、C P U 8 0 0 上で動作するソフトウェアが行うことになる。従って、処理の高速化を図れず、表示画面の全画素に対するガンマ補正を、1 フレーム内で完了するのは困難となる。また、C P U 8 0 0 の処理負荷が非常に重くなり、他の処理に悪影響を及ぼすという問題も招く。
- 10

- 一方、図 1 B の第 2 の手法では、専用のハードウェアであるガンマ補正回路 8 1 4 が使用されるため、高速なガンマ補正が可能になる。従って、表示画面の全画素に対するガンマ補正を 1 フレーム内で完了することも容易となる。また、C P U 8 1 0 の処理負荷も少ないため、他の処理に悪影響が及ぶ問題も解決できる。
- 15

しかしながら、図 1 B の第 2 の手法では、専用のハードウェアであるガンマ補正回路 8 1 4 が別途必要になってしまう。従って、ゲームシステムが大規模化し、製品コストの増加の問題を招く。

- 20 特に、家庭用ゲームシステムにおいては、製品の普及化を図るために、低コスト化が厳しく要求されており、ほとんどの家庭用ゲームシステムでは、図 1 B に示すようなガンマ補正回路がハードウェアとして設けられていない。従って、ガンマ補正を実現するためには、図 1 A のような第 1 の手法を採用せざるを得ない。

- 25 ところが、前述のように第 1 の手法では、1 フレーム内で全表示画面分のガンマ補正を完了するのは困難であり、他の処理にも悪影響を及ぼす。従って、家庭用ゲームシステムにおいては、ガンマ補正の実施自体を断念せざるを得な

かった。

また、従来のゲームシステムには以下のような課題もあった。

即ち、従来のゲームシステムにより生成される画像は、人間の視界画像のように視点からの距離に応じてフォーカシングされた画像ではなかった。このため、画像内の全ての被写体にピントが合っているかのような表現になっていた。

しかしながら、至近距離から遠距離までの全ての被写体にピントが合っている画像は、日常生活では見ることができない画像であるため、見た目に不自然さがあった。

よりリアリティを追求するためには、視点とオブジェクトとの距離や視線方向などに応じてピントの度合いが調節された画像を生成することが望ましい。しかしながら、ゲーム空間内の個々のオブジェクトについて視点との距離等を計算し、各オブジェクト毎にぼやけ具合を演算することで、ぼやけた画像を生成すると、処理負荷が過大になる。

リアルタイムに変化する視点に対応した画像を、制約されたハードウェア資源を用いて生成する必要があるゲームシステムにおいては、如何にして少ない処理負担で、現実世界の視界画像のようにフォーカシングされた画像を生成するかが重要な課題となる。

[発明の開示]

本発明は、以上のような課題に鑑みてなされたものであり、その目的とするところは、ガンマ補正などのビデオフィルタを少ない処理負担で実現できるゲームシステム、プログラム及び画像生成方法を提供することにある。

また本発明の他の目的は、よりリアルな画像を少ない処理負担で生成できるゲームシステム、プログラム及び画像生成方法を提供することにある。より具体的には、現実世界の視界画像のようにフォーカシングされた画像を、少ない処理負担で生成できるゲームシステム、プログラム及び画像生成方法を提供することにある。

上記課題を解決するために、本発明は、画像生成を行うゲームシステムであ

って、元画像の画像情報を、インデックスカラー・テクスチャマッピング用の
ルックアップテーブルのインデックス番号として設定する手段と、元画像の画
像情報がインデックス番号として設定された前記ルックアップテーブルを用い
て、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを
5 行い、元画像の画像情報を変換する手段とを含むことを特徴とする。また本発
明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であっ
て、上記手段をコンピュータに実現させるためのプログラムを含むことを特徴
とする。また本発明に係るプログラムは、コンピュータにより使用可能なプロ
グラム（搬送波に具現化されるプログラムを含む）であって、上記手段をコン
10 ピュータに実現させるための処理ルーチンを含むことを特徴とする。

本発明によれば、元画像の画像情報がインデックス番号として設定されたル
ックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラ
ー・テクスチャマッピングが行われ、元画像の画像情報を変換される。このよ
うに本発明によれば、ゲームシステム（画像生成システム）が元々有している
15 インデックスカラー・テクスチャマッピングの機能を有効利用して、元画像の
画像情報の変換が行われる。従って、ガンマ補正などのビデオフィルタの処理
を、新たなハードウェアを追加することなく、高速に実行できるようになり、
全表示画面分の画像情報を変換することも容易となる。

なお、元画像の画像情報は、例えば、描画領域（フレームバッファ、別バッ
20 ファ等）に描画されている情報であり、色情報、 α 値又は奥行き値などを含む。
また、元画像の画像情報の変換としては、ガンマ補正以外にも種々の変換を考
えることができる。また、元画像の画像情報をルックアップテーブルのインデ
ックス番号として設定することで得られる変換後の画像情報は、色情報に限定
されない。また、仮想オブジェクトとしては、1又は複数のプリミティブ面（ポ
25 リゴン、自由曲面）を考えることができる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記仮
想オブジェクトが、表示画面サイズのポリゴンであることを特徴とする。

このようにすれば、全表示画面分の元画像の画像情報を例えば1回（或いは数回）のテクスチャマッピングで変換できるようになる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とする。

このようにすれば、仮想オブジェクトを描画する領域の大きさを小さくすることが可能になり、記憶部の使用記憶容量を節約できる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換のための前記ルックアップテーブルを用意し、該ルックアップテーブルを用いて、元画像の画像情報に対してガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換を施すことを特徴とする。

このようにすれば、元画像に対して種々の画像エフェクトを施した画像を得ることが可能になり、生成される画像のバラエティ度を増すことができる。但し、本発明における画像情報の変換は、ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ、セピアフィルタに限定されるものではない。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像の画像情報が含む色情報の色成分を前記ルックアップテーブルのインデックス番号として設定して色情報を変換する場合において、変換後の色情報の他の色成分が描画領域に描画されないようにするためのマスク処理を行う手段（或いは該手段をコンピュータに実現させるためのプログラム又は処理ルーチン）を含むことを特徴とする。

このようにすれば、1つの入力値に対して複数の値が出力されてしまうインデックスカラー・テクスチャマッピング用のルックアップテーブルを用いながらも、1つの入力値に対して1つの値が出力されるガンマ補正などの画像変換

を実現できるようになる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像の画像情報が含む色情報の第Kの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Lの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Mの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報とを合成する手段（或いは該手段をコンピュータに実現させるためのプログラム又は処理ルーチン）を含むことを特徴とする。

このようにすれば、例えば第Kの色成分の入力に対して、第K、第L、第Mの色成分が合成された色情報の画像が生成されるような画像変換も実現できるようになる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像の画像情報の変換により、元画像の画像情報に応じた値の α 値が生成されることを特徴とする。

このようにすれば、例えば、元画像の画像情報が奥行き値である場合には、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定することが可能になる。これにより、例えば、元画像と該元画像に対応するぼかし画像とを、各画素に設定される α 値に基づいて合成することなどが可能になり、被写界深度の実現も可能になる。

また、生成された α 値を利用することで、元画像の画像情報の値に応じたマスク処理なども可能になる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像の画像情報が含む奥行き値が、前記ルックアップテーブルのインデックス番号として設定されることを特徴とする。

このように本発明においてルックアップテーブルのインデックス番号として設定される画像情報としては、種々の情報を考えることができる。

また本発明は、画像生成を行うゲームシステムであって、元画像の各画素の奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定する手段と、元画像の各画素の奥行き値がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定する手段と、元画像と該元画像に対応するぼかし画像とを、各画素に設定される α 値に基づいて合成する手段とを含むことを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であって、上記手段をコンピュータに実現させるためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、上記手段をコンピュータに実現させるための処理ルーチンを含むことを特徴とする。

本発明によれば、元画像の各画素の奥行き値がインデックス番号として設定されたルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングが行われ、元画像の各画素の奥行き値に応じた値に各画素の α 値が設定される。このように本発明によれば、ゲームシステム（画像生成システム）が元々有しているインデックスカラー・テクスチャマッピングの機能を有効利用して、奥行き値を α 値に変換できる。従って、奥行き値の変換処理を、例えば新たなハードウェアを追加することなく、高速に実行できるようになり、全表示画面分の奥行き値の変換も容易となる。

そして本発明によれば、元画像の各画素の奥行き値に応じた値に設定された α 値に基づいて、元画像とぼかし画像とが合成される。従って、奥行き値に応じて、ぼかし画像の合成比率等を変化させることが可能になり、被写界深度などの表現が可能になる。

なお、 α （アルファ）値は、各画素に関連づけられて記憶される情報であり、例えば色情報以外の情報である。また奥行き値は、視点（仮想カメラ）から近

いほど大きい値にしてもよいし、視点から遠いほど大きい値にしてもよい。また、元画像の合成対象となるぼかし画像の合成手法としては、種々の手法を採用できる。また α 値を用いた合成処理は α ブレンディングに限定されない。

5 また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像の各画素の奥行き値を、前記奥行き値の最上位ビットよりも下位のビット I ~ ビット J により構成される第 2 の奥行き値に変換し、前記第 2 の奥行き値を、インデックスカラー・テクスチャマッピング用の前記ルックアップテーブルのインデックス番号として設定することを特徴とする。

10 このようにすれば、 α 値の実効的な区分け数（奥行き値のしきい値の実効的な段階数）を増やすこと可能となり、例えば、仮想カメラの焦点（注視点）付近にあるオブジェクトのぼけ具合を、多段階のしきい値で精度良く制御できるようになる。従って、生成される画像の品質を高めることができる。

15 また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記奥行き値のビット I ~ ビット J 以外のビットの値に応じて、前記第 2 の奥行き値を所与の値にクランプすることを特徴とする。

このようにすれば、奥行き値のビット I ~ ビット J 以外のビットに 1 が立った場合等においても、矛盾の無い画像を生成できる。なお、所与の値としては、第 2 の奥行き値の最大値や最小値、或いは第 2 の奥行き値の上位のビット列を 1 にした値など、種々の値を考えることができる。

20 また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、前記ルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を前記第 2 の奥行き値に変換することを特徴とする。

25 このようにすれば、ルックアップテーブルの変換特性を変更するだけという少ない処理負担で、第 2 の奥行き値を所与の値にクランプするなどの多様な変換処理を実現できるようになる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記奥行き値のビットM～ビットNを、インデックスカラー・テクスチャマッピング用の第1のルックアップテーブルのインデックス番号として設定し、前記第1のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第3の奥行き値に変換し、
5 前記奥行き値のビットK～ビットL ($K \geq I \geq L > M \geq J \geq N$) を、インデックスカラー・テクスチャマッピング用の第2のルックアップテーブルのインデックス番号として設定し、前記第2のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第4の奥行き値に変換し、前記第3、第4の奥行き値に基づいて前記
10 第2の奥行き値を求めることを特徴とする。

このようにすれば、奥行き値の所定範囲のビット列（例えば0～7、8～15、16～23、24～31ビット）しか取り出せないような制限がある場合においても、奥行き値の任意のビットI～Jから構成される第2の奥行き値を得ることができる。これにより、 α 値を区分けする奥行き値のしきい値の実効的な段階数を増やすこと可能となり、生成される画像の品質を高めることができる。
15

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、元画像をテクスチャとして設定し、該テクスチャをテクセル補間方式で仮想オブジェクトにマッピングする際に仮想オブジェクトのテクスチャ座標をシフトさせ、
20 元画像のぼかし画像を生成することを特徴とする。

このようにすれば、テクスチャ座標をシフトさせながらテクセル補間方式で仮想オブジェクトに元画像をマッピングするだけという簡素な処理で、元画像のぼかし画像を生成できるようになる。

25 なお、テクセル補間方式とは、特には限定はされないが、テクセルの画像情報を補間してピクセルの画像情報を得る方式などであり、例えば、バイリニアフィルタ方式やトライリニアフィルタ方式などを考えることができる。

また、仮想オブジェクトは、ポリゴンなどのプリミティブ面であることが望ましいが、立体的なオブジェクトであってもよい。また、仮想オブジェクトは画面上に表示しないことが望ましいが、表示するようにしてもよい。

5 また、テクセル補間方式のテクスチャマッピングでぼかし画像を生成する場合には、テクスチャ座標を1テクセルよりも小さい値だけシフトさせることが望ましい。また、テクスチャ座標を第1のシフト方向へシフトさせてテクセル補間方式でテクスチャマッピングを行った後に、テクスチャ座標を第2のシフト方向にシフトさせてテクセル補間方式でテクスチャマッピングを行ってもよい。或いは、第1のシフト方向へのシフトと第2のシフト方向へのシフトのセ
10 ットを、複数回繰り返してもよい。

また本発明は、家庭用ゲームのゲーム画像を生成するためのゲームシステムであって、プレーヤがゲームコントローラを用いて入力した操作データに基づいて、モニタの表示特性を調整するための調整データを設定する手段と、設定された調整データを、プレーヤの個人データを記憶するためのセーブ用情報記憶装置にセーブするセーブ手段と、表示特性の調整により得られた調整データ
15 又はセーブ用情報記憶装置からロードされた調整データに基づいて、元画像の画像情報に対して変換処理を施す手段とを含むことを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であって、上記手段をコンピュータに実現させるためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、上記手段をコンピュータに実現させるための処理ルーチンを含むことを特徴とする。
20

本発明によれば、プレーヤは、ゲームコントローラを用いてモニタの表示特性（明るさ、色の濃さ、色合い、又はシャープネス等）を調整するための調整
25 データを設定できる。従って、プレーヤがモニタを直接操作しなくて済むため、プレーヤの利便性を向上できる。そして、設定された調整データはセーブ用情報記憶装置にセーブされると共に、表示特性の調整により得られた調整データ

又はセーブ用情報記憶装置からロードされた調整データに基づいて、元画像の画像情報の変換処理が行われる。従って、プレーヤがプレイするゲームに応じた最適な表示特性でゲーム画像を表示できるようになる。また、プレーヤが他の映像ソースの映像を見た場合などに、モニタの表示特性の調整の悪影響が及
5 んだりする事態を防止できる。

また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記セーブ手段が、画像情報の変換特性を表す自由曲線の制御点のデータを、前記調整データとしてセーブ用情報記憶装置にセーブすることを特徴とする。

このようにすれば、セーブ用情報記憶装置の使用記憶容量を節約できるようになり、空いた記憶容量を他の用途に使用できるようになる。
10

[図面の簡単な説明]

図 1 A、図 1 B は、ガンマ補正を実現する第 1、第 2 の手法について説明するための図である。

15 図 2 は、本実施形態のゲームシステムのブロック図の例である。

図 3 は、インデックスカラー・テクスチャマッピングについて説明するための図である。

図 4 は、インデックスカラー・テクスチャマッピング用の LUT を有効利用して、元画像を変換する手法について説明するための図である。

20 図 5 A、図 5 B は、本実施形態により生成されるゲーム画像の例である。

図 6 は、元画像を複数のブロックに分割し、各ブロックの画像を、LUT を用いて分割ブロックサイズのポリゴンにテクスチャマッピングする手法について説明するための図である。

図 7 A、図 7 B は、ガンマ補正の変換特性とガンマ補正用 LUT の例について示す図である。
25

図 8 A、図 8 B は、ネガポジ反転の変換特性とネガポジ反転用 LUT の例について示す図である。

図 9 A、図 9 B、図 9 C は、ポスタリゼーション、ソラリゼーション、2 値化の変換特性の例について示す図である。

図 10 A、図 10 B は、モノトーン（セピア）フィルタ用 LUT R、LUT G の例について示す図である。

5 図 11 は、モノトーン（セピア）フィルタ用 LUT B の例について示す図である。

図 12 は、ある色成分をインデックス番号に設定した場合に、他の色成分をマスクして、元画像の変換画像を得る手法について説明するための図である。

10 図 13 は、LUT R、LUT G、LUT B により得られた各色情報を合成して、元画像の変換画像を得る手法について説明するための図である。

図 14 は、LUT を利用したテクスチャマッピングを行い、 α プレーンを作成する手法について説明するための図である。

図 15 は、Z 値を LUT のインデックス番号に設定する手法について説明するための図である。

15 図 16 は、Z 値に応じた α 値を設定し、設定された α 値を用いて元画像とぼかし画像を合成する手法について説明するための図である。

図 17 A、図 17 B は、元画像とそのぼかし画像の例である。

図 18 は、Z 値に応じた α 値の設定手法について説明するための図である。

20 図 19 A、図 19 B、図 19 C は、仮想オブジェクトを描画することで仮想オブジェクトの奥側の画素の α 値を更新する手法について説明するための図である。

図 20 は、Z 値を Z2 値に変換し、その Z2 値を α 値に変換して元画像とぼかし画像を合成する手法について説明するための図である。

25 図 21 は、Z 値の最上位ビットを含む上位ビットで Z2 値を構成した場合の問題点について説明するための図である。

図 22 は、Z 値の最上位ビットよりも下位のビット I ~ J ビットで Z2 値を構成すると共に Z2 値を所与の値にクランプする手法について説明するための

図である。

図 2 3 は、L U T を用いて Z 値を Z 2 値に変換する手法について説明するための図である。

5 図 2 4 は、Z 値のビット 1 5 ~ 8 を変換する L U T 1 の具体例を示す図である。

図 2 5 は、Z 値のビット 2 3 ~ 1 6 を変換する L U T 2 の具体例を示す図である。

図 2 6 A、図 2 6 B は、Z 2 値を α 値に変換する L U T 3 の具体例とその変換特性曲線の例を示す図である。

10 図 2 7 は、クランプ処理について説明するための図である。

図 2 8 は、バイリニアフィルタ方式のテクスチャマッピングについて説明するための図である。

図 2 9 は、バイリニアフィルタ方式を有効利用してぼかし画像を生成する手法について説明するための図である。

15 図 3 0 は、バイリニアフィルタ方式を有効利用してぼかし画像を生成する手法について説明するための図である。

図 3 1 A、図 3 1 B は、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明するための図である。

20 図 3 2 A、図 3 2 B も、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明するための図である。

図 3 3 は、モニタの明るさ調整に関する従来の問題点について説明するための図である。

図 3 4 は、モニタの明るさ調整データをセーブ用情報記憶装置にセーブする手法について説明するための図である。

25 図 3 5 は、本実施形態の処理の詳細例について示すフローチャートである。

図 3 6 は、本実施形態の処理の詳細例について示すフローチャートである。

図 3 7 は、本実施形態の処理の詳細例について示すフローチャートである。

図 3 8 は、本実施形態の処理の詳細例について示すフローチャートである。

図 3 9 は、本実施形態の処理の詳細例について示すフローチャートである。

図 4 0 は、本実施形態を実現できるハードウェアの構成の一例を示す図である。

5 図 4 1 A、図 4 1 B、図 4 1 C は、本実施形態が適用される種々の形態のシステムの例を示す図である。

[発明を実施するための最良の形態]

以下、本発明の好適な実施形態について図面を用いて説明する。

10 1. 構成

図 2 に、本実施形態のゲームシステム（画像生成システム）のブロック図の一例を示す。なお同図において本実施形態は、少なくとも処理部 1 0 0 を含めばよく、それ以外のブロックについては、任意の構成要素とすることができる。

ここで処理部 1 0 0 は、システム全体の制御、システム内の各ブロックへの
15 命令の指示、ゲーム処理、画像処理、音処理などの各種の処理を行うものであり、その機能は、各種プロセッサ（CPU、DSP等）、或いはASIC（ゲートアレイ等）などのハードウェアや、所与のプログラム（ゲームプログラム）により実現できる。

20 操作部 1 6 0 は、プレーヤが操作データを入力するためのものであり、その機能は、レバー、ボタン、筐体などのハードウェアにより実現できる。

記憶部 1 7 0 は、処理部 1 0 0 や通信部 1 9 6 などのワーク領域となるもので、その機能はRAMなどのハードウェアにより実現できる。

25 情報記憶媒体（コンピュータにより使用可能な記憶媒体）1 8 0 は、プログラムやデータなどの情報を格納するものであり、その機能は、光ディスク（CD、DVD）、光磁気ディスク（MO）、磁気ディスク、ハードディスク、磁気テープ、或いはメモリ（ROM）などのハードウェアにより実現できる。処理部 1 0 0 は、この情報記憶媒体 1 8 0 に格納される情報に基づいて本発明（本

実施形態)の種々の処理を行う。即ち情報記憶媒体180には、本発明(本実施形態)の手段(特に処理部100に含まれるブロック)をコンピュータに実現させるための情報(プログラム或いはデータ)が格納される。

5 なお、情報記憶媒体180に格納される情報の一部又は全部は、システムへの電源投入時等に記憶部170に転送されることになる。また情報記憶媒体180に記憶される情報は、本発明の処理を行うためのプログラムコード、画像データ、音データ、表示物の形状データ、テーブルデータ、リストデータ、本発明の処理を指示するための情報、その指示に従って処理を行うための情報等の少なくとも1つを含むものである。

10 表示部190は、本実施形態により生成された画像を出力するものであり、その機能は、CRT、LCD、或いはHMD(ヘッドマウントディスプレイ)などのハードウェアにより実現できる。

 音出力部192は、本実施形態により生成された音を出力するものであり、その機能は、スピーカなどのハードウェアにより実現できる。

15 セーブ用情報記憶装置(携帯型情報記憶装置)194は、プレーヤの個人データ(セーブデータ)などが記憶されるものであり、このセーブ用情報記憶装置194としては、メモリカードや携帯型ゲーム装置などを考えることができる。

20 通信部196は、外部(例えばホスト装置や他のゲームシステム)との間で通信を行うための各種の制御を行うものであり、その機能は、各種プロセッサ、或いは通信用ASICなどのハードウェアや、プログラムなどにより実現できる。

25 なお本発明(本実施形態)の手段を実現(実行)するためのプログラム或いはデータは、ホスト装置(サーバー)が有する情報記憶媒体からネットワーク及び通信部196を介して情報記憶媒体180に配信するようにしてもよい。このようなホスト装置(サーバー)の情報記憶媒体の使用も本発明の範囲内に含まれる。

処理部 100 は、ゲーム処理部 110、画像生成部 130、音生成部 150 を含む。

ここでゲーム処理部 110 は、コイン（代価）の受け付け処理、各種モードの設定処理、ゲームの進行処理、選択画面の設定処理、オブジェクト（1 又は
5 複数のプリミティブ面）の位置や回転角度（X、Y 又は Z 軸回り回転角度）を
求める処理、オブジェクトを動作させる処理（モーション処理）、視点の位置
（仮想カメラの位置）や視線角度（仮想カメラの回転角度）を求める処理、マ
ップオブジェクトなどのオブジェクトをオブジェクト空間へ配置するための処
理、ヒットチェック処理、ゲーム結果（成果、成績）を演算する処理、複数の
10 プレーヤが共通のゲーム空間でプレイするための処理、或いはゲームオーバー
処理などの種々のゲーム処理を、操作部 160 からの操作データや、セーブ用
情報記憶装置 194 からの個人データや、ゲームプログラムなどに基づいて行
う。

画像生成部 130 は、ゲーム処理部 110 からの指示等にしたがって各種の
15 画像処理を行い、例えばオブジェクト空間内で仮想カメラ（視点）から見える
画像を生成して、表示部 190 に出力する。また、音生成部 150 は、ゲーム
処理部 110 からの指示等にしたがって各種の音処理を行い、BGM、効果音、
音声などの音を生成し、音出力部 192 に出力する。

なお、ゲーム処理部 110、画像生成部 130、音生成部 150 の機能は、
20 その全てをハードウェアにより実現してもよいし、その全てをプログラムによ
り実現してもよい。或いは、ハードウェアとプログラムの両方により実現して
もよい。

ゲーム処理部 110 は、移動・動作演算部 112、調整情報設定部 114、
セーブ部 116 を含む。

25 ここで移動・動作演算部 112 は、車などのオブジェクトの移動情報（位置
データ、回転角度データ）や動作情報（オブジェクトの各パーツの位置データ、
回転角度データ）を演算するものであり、例えば、操作部 160 によりプレー

ヤが入力した操作データやゲームプログラムなどに基づいて、オブジェクトを移動させたり動作させたりする処理を行う。

より具体的には、移動・動作演算部 112 は、オブジェクトの位置や回転角度を例えば 1 フレーム (1/60 秒) 毎に求める処理を行う。例えば (k-1) フレームでのオブジェクトの位置を PM_{k-1} 、速度を VM_{k-1} 、加速度を AM_{k-1} 、1 フレームの時間を Δt とする。すると k フレームでのオブジェクトの位置 PM_k 、速度 VM_k は例えば下式 (1)、(2) のように求められる。

$$PM_k = PM_{k-1} + VM_{k-1} \times \Delta t \quad (1)$$

$$VM_k = VM_{k-1} + AM_{k-1} \times \Delta t \quad (2)$$

調整情報設定部 114 は、プレーヤが操作部 160 (ゲームコントローラ) を用いて入力した操作データに基づいて、表示部 160 (モニタ) の表示特性 (明るさ、色の濃さ、色合い、又はシャープネス等) を調整するための調整データを設定 (作成) する処理を行う。

セーブ部 116 は、調整情報設定部 114 により設定された調整データ (明るさ、色の濃さ、色合い、又はシャープネス等を調整するためのデータ) を、セーブ用情報記憶装置 194 にセーブするための処理を行う。

そして本実施形態では、表示特性の調整により得られた調整データやセーブ用情報記憶装置 194 からロードされた調整データなどに基づいて、元画像の画像情報に対して変換処理が行われる。この場合の元画像の画像情報の変換処理は、インデックス番号設定部 134 や描画部 140 (テクスチャマッピング部 142) の機能により、実現されることになる。

画像生成部 130 は、ジオメトリ処理部 132、インデックス番号設定部 134、描画部 140 を含む。

ここで、ジオメトリ処理部 132 は、座標変換、クリッピング処理、透視変換、或いは光源計算などの種々のジオメトリ処理 (3 次元演算) を行う。そして、ジオメトリ処理後 (透視変換後) のオブジェクトデータ (オブジェクトの頂点座標などの形状データ、或いは頂点テクスチャ座標、輝度データ等) は、

記憶部 170 の主記憶領域 172 (メインメモリ) に保存される。

インデックス番号設定部 134 は、元画像の画像情報 (例えば透視変換後の画像の情報) を、LUT (ルックアップテーブル) 記憶部 178 に記憶されるインデックスカラー・テクスチャマッピング用の LUT のインデックス番号として設定するための処理を行う。ここで、元画像の画像情報としては、例えば、色情報 (RGB、YUV 等)、 α 値 (各画素に関連づけられて記憶される情報であり色情報以外のプラスアルファの情報)、奥行き値 (Z 値) 等、種々の情報を考えることができる。

そして本実施形態では、元画像の各画素の奥行き値がインデックス番号として設定されたルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行うことで、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定する。これにより、いわゆる被写界深度の表現が可能になる。

なお、仮想カメラの焦点から遠い画素ほど α 値が大きくなるように (広義には、ぼかし画像の合成比率が高くなるように)、LUT (ルックアップテーブル) における各画素の奥行き値 (インデックス番号) と α 値との対応関係を設定することが望ましい。また、LUT における各画素の奥行き値と α 値との対応関係を変化させることで、被写界深度の範囲やぼかしエフェクトの強弱を可変に制御してもよい。

描画部 140 は、ジオメトリ処理後のオブジェクト (モデル) を、描画領域 174 (フレームバッファ、別バッファ等のピクセル単位で画像情報を記憶できる領域) に描画するための処理を行うものであり、テクスチャマッピング部 142、 α 合成部 144、陰面消去部 146、マスク処理部 148 を含む。

ここでテクスチャマッピング部 142 は、テクスチャ記憶部 176 に記憶されるテクスチャをオブジェクトにマッピングするための処理 (オブジェクトにマッピングするテクスチャを指定する処理、テクスチャを転送する処理等) を行う。この場合、テクスチャマッピング部 142 は、LUT 記憶部 178 に記

憶されるインデックスカラー・テクスチャマッピング用のLUT（ルックアップテーブル）を用いたテクスチャマッピングを行うことができる。

そして本実施形態では、テクスチャマッピング部142が、元画像の画像情報がインデックス番号として設定されたLUTを用いて、仮想オブジェクト（表示画面サイズのポリゴン、分割ブロックサイズのポリゴン等）に対してテクスチャマッピングを行う。これにより、奥行き値（Z値）をNビット化する処理や、奥行き値を α 値に変換する処理や、種々の画像変換処理（ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ、セピアフィルタなどのビデオフィルタ）を少ない処理負担で実現できるようになる。

また本実施形態では、元画像と α 合成（ α ブレンディング、 α 加算、 α 減算、半透明処理等）されるぼかし画像（最もぼけた画像）の生成処理を、テクセル補間方式（バイリニアフィルタ方式、トライリニアフィルタ方式）のテクスチャマッピングを有効利用して実現している。

即ち、テクスチャマッピング部142は、テクスチャとして設定された元画像を、テクスチャ座標を例えば1ピクセル（テクセル）よりも小さい値だけシフトさせながら（例えば元画像の描画位置に基づき得られるテクスチャ座標からシフトさせながら）、テクセル補間方式で仮想オブジェクト（ぼかし領域と同一形状のオブジェクト）にマッピングする。このようにすれば、テクスチャ座標をシフトさせるだけという簡素な処理で、元画像の合成対象となるぼかし画像を生成できるようになる。

α 合成部144は、元画像とそのぼかし画像とを、描画領域174（フレームバッファ、別バッファ等）の各画素に対して設定された α 値（A値）に基づいて合成する処理を行う。例えば α 合成が α ブレンディングである場合には、下式のように元画像とぼかし画像とが合成される。

$$R_q = (1 - \alpha) \times R_1 + \alpha \times R_2 \quad (3)$$

$$G_q = (1 - \alpha) \times G_1 + \alpha \times G_2 \quad (4)$$

$$B_0 = (1 - \alpha) \times B_1 + \alpha \times B_2 \quad (5)$$

ここで、 R_1 、 G_1 、 B_1 は、描画領域174に既に描画されている元画像の色（輝度）のR、G、B成分であり、 R_2 、 G_2 、 B_2 は、ぼかし画像の色のR、G、B成分である。また R_0 、 G_0 、 B_0 は、 α ブレンディングにより生成されるR、G、B成分である。

また α 合成部144は、元画像の第Kの色成分（例えばR成分）をLUTのインデックス番号として設定することで得られる変換後の色情報（R、G、B）と、第Lの色成分（例えばG成分）をLUTのインデックス番号として設定することで得られる変換後の色情報と、第Mの色成分（例えばB成分）をLUTのインデックス番号として設定することで得られる変換後の色情報とを、合成するための処理（ α 加算、 α ブレンド等）も行う。

陰面消去部146は、Z値（奥行き値）が格納されるZバッファ179（奥行きバッファ、Zプレーン）を用いて、Zバッファ法のアルゴリズムにしたがった陰面消去を行う。本実施形態では、このZバッファ179に書き込まれたZ値を α 値に変換し、その α 値に基づいて元画像とぼかし画像の合成処理を行っている。

マスク処理部148は、元画像の色成分（例えばR成分）をLUTのインデックス番号として設定して色情報を変換する場合において、変換後の色情報の他の色成分（例えばG、B成分）が描画領域（フレームバッファ或いは別バッファ）に描画されないようにするためのマスク処理を行う。なお、G成分をインデックス番号として設定して色情報を変換する場合には、R、B成分に対するマスク処理を行い、B成分をインデックス番号として設定して色情報を変換する場合には、R、G成分に対するマスク処理を行うことになる。

なお、本実施形態のゲームシステムは、1人のプレーヤのみがプレイできるシングルプレーヤモード専用のシステムにしてもよいし、このようなシングルプレーヤモードのみならず、複数のプレーヤがプレイできるマルチプレーヤモードも備えるシステムにしてもよい。

また複数のプレーヤがプレイする場合に、これらの複数のプレーヤに提供するゲーム画像やゲーム音を、1つの端末を用いて生成してもよいし、ネットワーク（伝送ライン、通信回線）などで接続された複数の端末（ゲーム装置、携帯電話）を用いて生成してもよい。

5 2. 本実施形態の特徴

2. 1 インデックスカラー・テクスチャマッピングの利用

さて、前述のように、図1Aに示す第1の手法では、CPUの処理負荷が過大になってしまうため、ガンマ補正（ビデオフィルタ）を実現するのは実質的に不可能である。また図1Bに示す第2の手法では、専用のハードウェアであるガンマ補正回路が別途必要であるため、このようなガンマ補正回路が設けられていない家庭用ゲームシステムでは、ガンマ補正を実現できない。

そこで、本出願の発明者は、インデックスカラー・テクスチャマッピングにおいて使用されるルックアップテーブルLUTの存在に着目した。

即ち、インデックスカラーテクスチャマッピングでは、テクスチャ記憶部の使用記憶容量を節約するために、図3のA1に示すように、実際の色情報（RGB）ではなくインデックス番号が、テクスチャの各テクセルに関連づけて記憶される。また、図3のA2に示すように、インデックスカラー・テクスチャマッピング用のLUT（カラーパレット）には、インデックス番号により指定される色情報が記憶される。そして、オブジェクトに対してテクスチャマッピングを行う際には、テクスチャの各テクセルのインデックス番号に基づいてLUTを参照し、対応する色情報をLUTから読み出し、読み出された色情報をフレームバッファに描画する。

このようなインデックスカラーモードのテクスチャマッピングでは、LUTを用いない通常モードのテクスチャマッピングに比べて、使用できる色数は少なくなる（例えば256色）。しかしながら、テクスチャ記憶部に実際の色情報（例えば16ビットの色情報）を記憶する必要が無くなるため、テクスチャ記憶部の使用記憶容量を大幅に節約できる。

本実施形態は、このようなインデックスカラー・テクスチャマッピングを通常とは異なる形態で利用している点に特徴がある。

即ち、まず図4のB1に示すように、フレームバッファ（広義には描画領域）に描画されている元画像の各画素の画像情報（例えば色情報）を、ガンマ補正用のルックアップテーブルLUTのインデックス番号として設定する（インデックス番号とみなす）。そしてB2に示すように、元画像の画像情報がインデックス番号として設定されたLUTを用いて、仮想オブジェクト（例えば表示画面サイズのポリゴン）に対してインデックスカラー・テクスチャマッピングを行い、元画像の画像情報を変換する。そしてB3に示すように、変換後の画像情報を、フレームバッファ（描画領域）などに描き戻す。

以上のようにして本実施形態では、図5Aに示すような元画像から、図5Bに示すようなガンマ補正が施された画像を得ることに成功している。即ち図5Bの画像では、図5Aに比べて、よりコントラストのはっきりした画像になっている。

例えば図1Aの第1の手法では、元画像の色情報の読み出し、ガンマ補正用LUTの参照、色情報のフレームバッファへの書き戻しなどの全ての処理をCPU上で動作するソフトウェアが行うことになるため、処理の高速化を図れないと共に、CPUの処理負荷も過大になる。

これに対して本実施形態では、インデックスカラー・テクスチャマッピングを有効利用してガンマ補正を実現しており、このインデックスカラー・テクスチャマッピングは、専用のハードウェアである描画プロセッサ（描画部）により高速に実行される。従って本実施形態によれば、図1Aの第1の手法に比べて高速にガンマ補正を実行でき、全表示画面分のガンマ補正を1フレーム（例えば1/60秒、1/30秒）内で完了することも容易となる。

また、インデックスカラー・テクスチャマッピングは、メインプロセッサ（CPU）とは独立に動作する描画プロセッサにより実行できるため、メインプロセッサ（CPU）の処理負荷の増加も最小限に抑えることができる。従って、

ガンマ補正の実行が要因となって他の処理に悪影響が及ぶ事態も防止できる。

また、従来のゲームシステムでは、描画プロセッサの処理能力はそれほど高くなかった。従って、フレームバッファへの元画像の描画と、表示画面サイズのポリゴンの描画を、1フレーム内で完了させることは難しかった。

5 しかしながら、ゲームシステムにおいて、描画プロセッサの処理能力の向上は、他の回路ブロックの処理能力の向上に比べて著しく大きく、非常に高いフィルレート（1秒間にレンダリングできるテクセル数）を持つ描画プロセッサがゲームシステムに使用されるようになってきた。従って、フレームバッファへの元画像の描画と、表示画面サイズのポリゴンの描画を、1フレーム内で完了させることも容易となり、インデックスカラー・テクスチャマッピングを有効利用したガンマ補正も無理なく実現できるようになってきた。

10

また図1Bの第2の手法では、専用のハードウェアであるガンマ補正回路が別途必要になり、ゲームシステムの高コスト化を招く。また、このようなガンマ補正回路を元々有しない家庭用ゲームシステムなどでは、図1Bに示す第2の手法を実現することはできず、図1Aの手法を採用せざるを得なかった。

15

これに対して本実施形態では、インデックスカラー・テクスチャマッピングを有効利用してガンマ補正を実現しており、このインデックスカラー・テクスチャマッピングは、描画プロセッサが元々持っているハードウェアにより実行される。従って本実施形態によれば、図1Bのようなガンマ補正回路を新たに付加する必要がなく、ゲームシステムが高コスト化してしまう事態を防止できる。また、ガンマ補正回路を元々有していない家庭用ゲームシステムにおいても、ハードウェアによる高速なガンマ補正を実現できるようになる。

20

なお、図4では表示画面サイズのポリゴンにテクスチャマッピングしてガンマ補正（ビデオフィルタ）を実現しているが、表示画面を分割したブロックのサイズのポリゴンにテクスチャマッピングするようにしてもよい。

25

即ち、図6のC1に示すように、フレームバッファ上の元画像（表示画面）を複数のブロックに分割し、C2に示すように、各ブロックの画像を、LUT

を用いて分割ブロックサイズのポリゴンにテクスチャマッピングする。そして、得られた分割ブロックサイズの画像をフレームバッファ（描画領域）に描き戻す。

5 或いは、透視変換後（スクリーン座標系への変換後）のオブジェクトの画像の全部又は一部を内包し、透視変換後のオブジェクトの大きさに応じてその大きさが変化するようなポリゴン（仮想オブジェクト）を生成し、そのポリゴンにテクスチャマッピングする。

10 このようにすれば、例えばテクスチャマッピングされたポリゴンを別バッファに一時的に描画するような場合に、VRAM上での別バッファの占有領域を小さくできる。

即ち、図4のように表示画面サイズのポリゴンにテクスチャマッピングすると、この表示画面サイズのポリゴンを一時的に描画するために、表示画面サイズの別バッファをVRAM上に確保しなければならず、他の処理に支障を来すおそれがある。

15 図6のように、分割ブロックサイズのポリゴンにテクスチャマッピングするようにすれば、VRAM上には分割ブロックサイズの別バッファを用意すれば済むため、別バッファの占有領域を小さくできる。従って、限られたハードウェア資源を有効利用することが可能になる。

2. 2 各種のビデオフィルタ（LUT）の例

20 図7Aにガンマ補正の変換特性例を示す。

図7Aでは4つの制御点CP0、CP1、CP2、CP3により、ガンマ補正の変換特性を表すベジエ曲線（広義には自由曲線）が特定される。この場合に、CP0のY座標は $Y_0 = 0$ に設定され、CP3のY座標は $Y_3 = 255$ に設定される。そして、CP1、CP2のY座標である Y_1 、 Y_2 を可変に制御
25 することで、ガンマ補正の変換特性を調整できる。

そして、ガンマ補正における入力値Xと出力値Yの関係式は、例えば下式のようなになる。

$$Y = Y_{20} + (X / 255) \times (Y_{21} - Y_{20}) \quad (3)$$

但し、

$$Y_{20} = Y_{10} + (X / 255) \times (Y_{11} - Y_{10})$$

$$Y_{21} = Y_{11} + (X / 255) \times (Y_{12} - Y_{11})$$

$$5 \quad Y_{10} = Y_0 + (X / 255) \times (Y_1 - Y_0)$$

$$Y_{11} = Y_1 + (X / 255) \times (Y_2 - Y_1)$$

$$Y_{12} = Y_2 + (X / 255) \times (Y_3 - Y_2)$$

である。

10 上式(3)の入力値Xにインデックス番号を設定し、出力値Yに各色成分の出力ROUT、GOUT、BOUTを設定することで、図7Bに示すようなガンマ補正用LUTを作成する。そして、作成されたLUTをVRAMに転送し、このLUTを用いて図4等で説明したインデックスカラー・テクスチャマッピングを行うことで、元画像に、ガンマ補正のビデオフィルタを施した画像を得ることができる。

15 さて、本実施形態によれば、フレームバッファ上の元画像に対して施すビデオフィルタとして、ガンマ補正以外にも種々のビデオフィルタを実現できる。

図8Aに、ネガポジ反転のビデオフィルタの変換特性例を示す。ネガポジ反転における入力値Xと出力値Yの関係式は下式のようにになる。

$$Y = 255 - X \quad (4)$$

20 上式(4)の入力値Xにインデックス番号を設定し、出力値Yに各色成分の出力ROUT、GOUT、BOUTを設定することで、図8Bに示すようなネガポジ反転用LUTを作成する。そして、このLUTを用いて図4で説明したインデックスカラー・テクスチャマッピングを行うことで、元画像に、ネガポジ反転のビデオフィルタを施した画像を得ることができる。

25 図9Aに、多階調の画像をいくつかの階調に制限して表示するためのポスタリゼーションのビデオフィルタの変換特性例を示す。ポスタリゼーションにおける入力値Xと出力値Yの関係式は下式のようにになる。

$$Y = \{ \text{INT} (X / \text{VAL}) \} \times \text{VAL} \quad (5)$$

上式において $\text{INT} (R)$ は、 R の小数点を切り捨てて整数化する関数であり、 VAL は任意の値である。

5 なお、図 9 B には、ある点において入力値と出力値の曲線関数の傾きが反転するような画像効果を奏するソラリゼーションのビデオフィルタの変換特性例を示す。また、図 9 C には、画像のハイコントラスト効果を実現する 2 値化のビデオフィルタの変換特性例を示す。

また本実施形態によれば、モノトーンフィルタやセピアフィルタのビデオフィルタも実現できる。

10 モノトーンフィルタの変換前の色成分を RIN 、 GIN 、 BIN とし、変換後の色成分を $ROUT$ 、 $GOUT$ 、 $BOUT$ とした場合に、モノトーンフィルタの変換式は例えば下式のようにになる。

$$ROUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN \quad (6)$$

$$GOUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN \quad (7)$$

15 $BOUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN \quad (8)$

ここで、入力値 RIN に対する出力値 ($ROUTR$ 、 $GOUTR$ 、 $BOUTR$) の関係式として下式 (9) を、入力値 GIN に対する出力値 ($ROUTG$ 、 $GOUTG$ 、 $BOUTG$) の関係式として下式 (10) を、入力値 BIN に対する出力値 ($ROUTB$ 、 $GOUTB$ 、 $BOUTB$) の関係式として下式 (11) を定義する。

$$\begin{aligned} & (ROUTR, GOUTR, BOUTR) \\ & = (0.299 \times RIN, 0.299 \times RIN, 0.299 \times RIN) \end{aligned} \quad (9)$$

$$\begin{aligned} & (ROUTG, GOUTG, BOUTG) \\ & = (0.587 \times GIN, 0.587 \times GIN, 0.587 \times GIN) \end{aligned} \quad (10)$$

25
$$\begin{aligned} & (ROUTB, GOUTB, BOUTB) \\ & = (0.114 \times BIN, 0.114 \times BIN, 0.114 \times BIN) \end{aligned} \quad (11)$$

上式 (9)、(10)、(11) に基づいて、図 10 A、図 10 B、図 11

に示すようなモノトーンフィルタ用のルックアップテーブルLUTR、LUTG、LUTBを作成する。そして、これらのLUTR、LUTG、LUTBを用いてインデックスカラー・テクスチャマッピングを行うことで、元画像にモノトーンフィルタを施した画像を得ることができる。

5 なお、セピアフィルタの場合の変換式は例えば下式のようなになる。

$$ROUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN + 6 \quad (12)$$

$$GOUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN - 3 \quad (13)$$

$$BOUT = 0.299 \times RIN + 0.587 \times GIN + 0.114 \times BIN - 3 \quad (14)$$

但し、 $0 \leq (ROUT, GOUT, BOUT) \leq 255$ となるようにクラン

10 プ処理を行う。

そして、セピアフィルタの場合は例えば下記のような式を定義できる。

$$(ROUTR, GOUTR, BOUTR)$$

$$= (0.299 \times RIN + 2, 0.299 \times RIN - 1, 0.299 \times RIN - 1) \quad (15)$$

$$(ROUTG, GOUTG, BOUTG)$$

$$15 \quad = (0.587 \times GIN + 2, 0.587 \times GIN - 1, 0.587 \times GIN - 1) \quad (16)$$

$$(ROUTB, GOUTB, BOUTB)$$

$$= (0.114 \times BIN + 2, 0.114 \times BIN - 1, 0.114 \times BIN - 1) \quad (17)$$

そして、上式(15)、(16)、(17)に基づいて、セピアフィルタ用のルックアップテーブルLUTR、LUTG、LUTBを作成する。そして、
20 これらのLUTR、LUTG、LUTBを用いてインデックスカラー・テクスチャマッピングを行うことで、元画像にセピアフィルタを施した画像を得ることができる。

2. 3 マスク処理

さて、ガンマ補正では、1つの入力値(RIN、GIN又はBIN)に対して1つの値(ROUT、GOUT又はBOUT)が出力されるようなLUTが必要になる。
25

ところが、図3に示すインデックスカラー・テクスチャマッピング用のLU

Tは、元々、ガンマ補正用に設計されたものではないため、1つ入力値（インデックス番号）に対して、複数の値（例えばROUT、GOUT、及びBOUT）が出力されてしまう。従って、このLUTの不整合を解決しなければならないという課題がある。

- 5 そこで本実施形態では、元画像の画像情報（R、G、B、Z値又は α 値等）をLUTのインデックス番号として設定する場合において、変換により得られた画像情報のうち必要な画像情報のみが描画領域（フレームバッファ、別バッファ）に描画され、他の画像情報が描画されないようにするためのマスク処理を行っている。元画像の画像情報が色情報である場合を例にとれば、元画像の
- 10 1つの色成分をLUTのインデックス番号として設定する場合において、変換により得られる他の色成分が描画領域に描画されないようにするためのマスク処理を行っている。

- より具体的には、図12のD1に示すように、元画像のRプレーンの値をインデックス番号に設定して、LUTを用いたテクスチャマッピングを行うと、
- 15 R（ROUT）、G（GOUT）、B（BOUT）という3つプレーンの値が出力される。そして、この場合には、D2に示すように、出力されたRプレーンの値のみを描画領域に描画し、Gプレーン、Bプレーンの値についてはマスクして描画領域に描画されないようにする。

- また、図12のD3に示すように、元画像のGプレーンの値をインデックス番号に設定してテクスチャマッピングを行った場合には、D4に示すように、
- 20 出力されたGプレーンの値のみを描画領域に描画し、Rプレーン、Bプレーンの値についてはマスクして描画領域に描画されないようにする。

- また、図12のD5に示すように、元画像のBプレーンの値をインデックス番号に設定してテクスチャマッピングを行った場合には、出力されたBプレー
- 25 ンの値のみを描画領域に描画し、Rプレーン、Gプレーンの値についてはマスクして描画領域に描画されないようにする。

 以上のようにすることで、元々はガンマ補正用に設計されていないインデッ

クスカラー・テクスチャマッピング用のLUTを用いながらも、少ない処理負荷で、元画像の画像変換処理を実行できるようになる。

2. 4 合成

図12で説明したマスク処理の手法は、ガンマ補正の他にも、図8A～図9Cにおいて説明したネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化などのビデオフィルタなどに有効である。

これに対して、モノトーンフィルタやセピアフィルタなどのビデオフィルタを実現する場合には、次のような手法を採用することが望ましい。即ち、元画像のR成分をLUTRのインデックス番号として設定することで得られる色情報(R、G、B)と、G成分をLUTGのインデックス番号として設定することで得られる色情報(R、G、B)と、B成分をLUTBのインデックス番号として設定することで得られる色情報(R、G、B)とを合成するようにする。

より具体的には、図13のE1に示すように、元画像のR(RIN)プレーンの値をインデックス番号に設定して、図10AのLUTRを用いたテクスチャマッピングを行うことで、E2に示すようなR(ROUTR)、G(GOUTR)、B(BOUTR)という3つプレーンの値を得る。この場合の、RINと(ROUTR、GOUTR、BOUTR)の関係式は上式(9)又は(15)のようになっている。

また、図13のE3に示すように、元画像のG(GIN)プレーンの値をインデックス番号に設定して、図10BのLUTGを用いたテクスチャマッピングを行うことで、E4に示すようなR(ROUTG)、G(GOUTG)、B(BOUTG)という3つプレーンの値を得る。この場合の、GINと(ROUTG、GOUTG、BOUTG)の関係式は上式(10)又は(16)のようになっている。

また、図13のE5に示すように、元画像のB(BIN)プレーンの値をインデックス番号に設定して、図11のLUTBを用いたテクスチャマッピングを行うことで、E6に示すようなR(ROUTB)、G(GOUTB)、B(B

OUTB) という3つプレーンの値を得る。この場合の、BINと(ROUTB、GOUTB、BOUTB)の関係式は上式(11)又は(17)のようになっている。

そして、図13のE7に示すように、E2に示すR(ROUTR)、G(GOUTR)、B(BOUTR)の色情報と、E4に示すR(ROUTG)、G(GOUTG)、B(BOUTG)の色情報と、E6に示すR(ROUTB)、G(GOUTB)、B(BOUTB)の色情報とを、合成(加算)する。

このようにすることで、上式(6)、(7)、(8)或いは(12)、(13)、(14)の変換式に示すような、モノトーンフィルタ或いはセピアフィルタのビデオフィルタを実現できるようになる。

2. 5 Z値、 α 値への利用

以上では、インデックスカラー・テクスチャマッピング用のLUTに基づき出力される色情報R、G、Bを利用する場合について説明した。

しかしながら、インデックスカラー・テクスチャマッピング用のLUTに基づき出力される α 値(A値。画素に関連づけて設定される色情報以外の情報)を利用するようにしてもよい。

例えば図14に示すように、Rプレーン(又はG、B)の値をインデックス番号として設定して、LUTを用いたテクスチャマッピングを行い、 α (α OUT)プレーンを生成する。そして、生成された α プレーンを用いて、マスク処理などを行うようにする。

即ち、例えばR値が0~127の時には α 値(α OUT)が0になり、R値が128~255の時には α 値が255になるように α 値が設定されたLUTを使用する。そして、 α 値が255よりも小さい画素についてはマスク処理を行わず、 α 値が255の画素についてはマスク処理を行うようにする。このようにすれば、R値が128以上の画素に対してだけマスク処理が行われるようになり、各画素のR値の大小に応じたマスク処理を行うことができるようになる。

なお、生成された α プレーンの値を α 合成の係数（透明度、半透明度、不透明度）として使用してもよい。

また、LUTのインデックス番号として設定される画像情報は色情報に限定されない。即ち、描画領域（VRAM）上にあり、LUTのインデックス番号として設定できる画像情報であればよい。

例えば図15に示すように、Z値（奥行き値）をLUTのインデックス番号として設定するようにしてもよい。

そして、この場合には、Z値をインデックス番号に設定してインデックスカラー・テクスチャマッピングを行うことで得られる α プレーンの値を、例えば α 合成の係数として使用するようにする。このようにすることで、Z値に応じた値の α 値を設定できるようになり、ぼかし画像を用いた被写界深度などの表現が可能になる。

即ち図15に示すようなLUTを用いたテクスチャマッピングを行うことで、図16のF1に示すように、元画像の各画素A、B、C、DのZ値 Z_A 、 Z_B 、 Z_C 、 Z_D に応じた値に各画素の α 値 α_A 、 α_B 、 α_C 、 α_D を設定する。そして、例えば図16のF2に示すような α プレーンを生成する。より具体的には、仮想カメラ10の焦点（注視点）から遠い画素（焦点とのZ値の差が大きい画素）ほど、例えば大きな α 値を設定する。これにより、仮想カメラ10の焦点から遠い画素ほど、ぼかし画像の合成比率が高くなる。

そして、図16のF3に示すように、生成された α プレーン（各画素に設定された α 値）に基づいて、元画像とぼかし画像の α 合成（ α ブレンディング等）を行う。なお、図17Aに元画像の例を示し、図17Bに、そのぼかし画像の例を示す。

このように、Z値（奥行き値）に応じて設定された α 値に基づき元画像（図17A）とぼかし画像（図17B）の α 合成を行うことで、例えば、仮想カメラの焦点（ピントが合っている点として設定される点）から遠くなるほどぼけて見える画像を生成できるようになり、いわゆる被写界深度の表現が可能にな

る。これにより、画面内の全ての被写体にピントが合っていた従来のゲーム画像とは異なり、現実世界の視界画像のように視点からの距離に応じてフォーカシングされたリアルで自然なゲーム画像を生成できる。この結果、プレーヤの仮想現実感を格段に向上できる。

- 5 例えば図18に、Z値に応じた α 値の設定の一例を示す。なお、図18では、 α 値は、その大きさが1.0以下になるように正規化されている。

そして図18では、Z値Z1～Z4、Z1'～Z4'（しきい値）により領域AR0～AR4、AR1'～AR4'の区分けが行われる。そして、これらの領域AR0～AR4、AR1'～AR4'に対して、 α 値 $\alpha_0 \sim \alpha_4$ 、 $\alpha_1' \sim \alpha_4'$ が設定される。

例えば、Z1～Z2の間の領域AR1にある画素については、その α 値が α_1 に設定され、Z2～Z3の間の領域AR2にある画素については、その α 値が α_2 に設定される。また、Z1'～Z2'の間の領域AR1'にある画素については、その α 値が α_1' に設定され、Z2'～Z3'の間の領域AR2'にある画素については、その α 値が α_2' に設定される。

そして、各領域に設定される α 値には例えば以下の関係式が成り立つ。

$$\alpha_0 < \alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 \quad (18)$$

$$\alpha_0 < \alpha_1' < \alpha_2' < \alpha_3' < \alpha_4' \quad (19)$$

これらの式(18)、(19)から明らかなように、仮想カメラ10の焦点（注視点）から遠いほど α 値が大きくなっている。即ち、仮想カメラ10の焦点とのZ値の差が大きい画素ほど、ぼかし画像の合成比率が高くなるように、 α 値が設定されている。

このように α 値を設定することで、仮想カメラの焦点から遠くなるほどぼけて見える画像を生成でき、いわゆる被写界深度の表現が可能になる。

25 しかも本実施形態では、LUTを用いた1回のテクスチャマッピングで、各画素のZ値を α 値に変換できるため、処理負担が非常に軽いという利点がある。

即ち、LUTを用いない α 値の設定手法の例として、図19A、図19B、

図 1 9 C に示す手法が考えられる。

この手法では図 1 9 A に示すように、まず、Z 値が Z 1 に設定される仮想オブジェクト O B 1 (ポリゴン) を、フレームバッファに描画することで、仮想オブジェクト O B 1 を基準に奥側にある画素の α 値を更新する。即ち、Z 値に
5 基づく陰面消去手法を有効利用して、O B 1 よりも奥側にある画素の α 値を更新する。

次に、図 1 9 B に示すように、Z 値が Z 2 に設定される仮想オブジェクト O B 2 をフレームバッファに描画することで、仮想オブジェクト O B 2 を基準に奥側にある画素の α 値を更新する。同様に、図 1 9 C に示すように、Z 値が Z
10 3 に設定される仮想オブジェクト O B 3 をフレームバッファに描画することで、仮想オブジェクト O B 3 を基準に奥側にある画素の α 値を更新する。

このような手法を採用すれば、領域 A R 1 にある画素の α 値を $\alpha 1$ に設定し、領域 A R 2 にある画素の α 値を $\alpha 2$ に設定し、領域 A R 3 にある画素の α 値を $\alpha 3$ に設定できるようになる。即ち、L U T を用いなくても、各画素の Z 値に
15 応じた値に各画素の α 値を設定できる。

しかしながら、この手法では、Z 値のしきい値の段階数分だけ、仮想オブジェクトの描画処理が必要になる。例えば図 1 8 の場合は 8 回の描画処理が必要になる。従って、この手法には描画処理の負担が重くなるという欠点がある。一方、描画処理の負担を軽減するために Z 値のしきい値の段階数を減らすと、
20 今度は、Z 値のしきい値の境界が表示画面上で帯状に見えてしまう事態が生じ、表示品質が低下する。

L U T を用いて Z 値を α 値に変換する本実施形態の手法によれば、L U T を用いた 1 回のテクスチャマッピングで、各画素の Z 値を α 値に一斉に変換できる。例えば、L U T のインデックス番号 (エントリー) のビット数が 8 ビット
25 の場合には、L U T を用いた 1 回のテクスチャマッピングで、2 5 6 段階の Z 値のしきい値で区分けされる α 値を得ることができる。従って、Z 値のしきい値の境界が表示画面上で帯状に見えてしまう事態を防止でき、少ない処理負担

で高品質な画像を生成できる。

2. 6 Z値の8ビット化

さて、ゲームシステムにおいては、陰面消去の精度を高めるために、Z値のビット数は例えば24ビット、32ビットというように非常に多い。

- 5 一方、インデックスカラー・テクスチャマッピング用のLUTのインデックス番号のビット数（LUTのエントリーのビット数）は、例えば8ビットというようにZ値のビット数に比べて少ない。

- 従って、図15のようにインデックスカラー・テクスチャマッピング用のLUTを用いてZ値を α 値に変換する場合には、その前処理として、Z値を、LUTのインデックス番号のビット数と同じビット数のZ値（以下、Z2値と呼ぶ）に変換する処理が必要になる。即ち、LUTのインデックス番号のビット数が8ビットである場合には、Z値を8ビットのZ2値（第2の奥行き値）に変換する処理が必要になる。
- 10

- この場合に、矛盾の無いZ2値を得るためには、Z値の最上位ビットを含む上位の8ビットをビットセレクトしたものをZ2値とする必要がある。即ち図20において、ビット23（最上位ビット）～16の8ビットをビットセレクトして、Z2値として設定する。
- 15

- しかしながら、このようにZ値の上位ビット23～16の8ビットをZ2値に設定し、このZ2値をLUTを用いて α 値に変換すると、 α 値の区分け数が少なくなってしまうということが判明した。
- 20

例えば図21のように、Z値が4ビットであり、Z値の上位の2ビットをセレクトしてZ2値に設定し、このZ2値を α 値に変換した場合を考える。この場合には、Z = 0～15の全範囲において4段階のしきい値で α 値が区分けされることになる。

- 25 しかしながら、図21のOB1のようにスクリーンSC（透視変換面）の手前にあるオブジェクトについては、通常はニアクリップされてしまう。そして、例えば図21においてZ = 10～15の範囲にあるオブジェクトがニアクリッ

プされるような場合には、Z値の最上位ビットに1が立つことは希となる。また、OB2のようにスクリーンSCの近くにあるオブジェクトについては、何れにせよ最もぼけた画像になるため、ぼけ具合を精度良く制御する必要がない。従って、図21のG1に示す部分における2段階の区分けは無駄になる。

- 5 そこで本実施形態では図20に示すように、元画像の各画素のZ値（奥行き値）を、Z値の最上位ビットよりも下位のビットI～J（例えばビット19～12）により構成されるZ2値（第2の奥行き値）に変換し、このZ2値を、インデックスカラー・テクスチャマッピング用のLUTのインデックス番号として設定してテクスチャマッピングを行い、各画素の α 値を求める。そして、
10 この求められた各画素の α 値に基づいて、元画像とぼかし画像を α 合成する。

 このようにすれば図22に示すように、仮想カメラ10の焦点（注視点）付近にあるオブジェクト（例えばOB3、OB4）についてだけ、多段階（図22では4段階）のしきい値で区分けされた α 値により、ぼけ具合を精度良く制御できるようになる。従って、生成される画像の品質を高めることができる。

- 15 そして本実施形態では、Z値のビットI～J以外のビットの値に応じて、Z2値を所与の値にクランプするようにしている。より具体的には図22に示すように、Z値の上位のビットに例えば1が立った場合には、Z2値を最大値（広義には所与の値）にクランプする。このようにすれば、OB1やOB2のように、ぼけ具合を精度良く制御する必要がないオブジェクトについては、Z2値
20 が最大値に設定され、最もぼけた画像に設定される。従って、Z値を、そのビットI～Jにより構成されるZ2値に変換しても、矛盾の無い画像を生成できる。

2. 7 LUTを利用したZ値の変換処理

- さて本実施形態では、図20で説明したZ値をZ2値に変換する処理（8ビット化処理）を、インデックスカラー・テクスチャマッピング用のLUTを利用したテクスチャマッピングにより実現している。即ち、Z値をLUTのイン
25 デックス番号として設定し、そのLUTを用いて仮想オブジェクトに対してイ

ンデックスカラー・テクスチャマッピングを行い、Z値をZ₂値に変換する。

例えば図23に示すように24ビットのZ値をLUTを用いて変換する場合を考える。この場合には図23のH1に示すように、Z値のビット15～8（ビットM～N）をLUT1（第1のルックアップテーブル）のインデックス番号に設定し、LUT1を用いてインデックスカラー・テクスチャマッピングを行い、Z値をZ₃値（第3の奥行き値）に変換する。

次に図23のH2に示すように、Z値のビット23～16（ビットK～L）をLUT2（第2のルックアップテーブル）のインデックス番号に設定し、LUT2を用いてインデックスカラー・テクスチャマッピングを行い、Z値をZ₄値（第4の奥行き値）に変換する。

そして図23のH4に示すようにZ₃値とZ₄値とに基づいてZ₂値を求め、H5に示すようにこのZ₂値をLUT3（第3のルックアップテーブル）を用いて α 値に変換する。

より具体的には、LUT1の変換により得られたZ₃値を描画領域（フレームバッファ、別バッファ）に描画する。その後、LUT2の変換により得られたZ₄値を描画領域に描画する。この際、Z₃値の下位の4ビット（有効ビット）をマスクして、これらのビットにZ₄値が上書きされないようにして、Z₂値を求める。

図23の手法を採用することで、Z値の任意の8ビット（広義には任意のビットI～J）を取り出すことが可能になる。

即ち、LUTのインデックス番号に設定するためにZ値の8ビットを取り出そうとした場合に、Z値のビット23～16、15～8又は7～0というように所定範囲の8ビットしか取り出せない場合がある。

一方、Z値の中のどの8ビットを取り出すかは、図20、図21で説明したように、ニアクリップの範囲や、仮想カメラの焦点（注視点）の位置などに応じて決められる。

従って、Z値のビット23～16、15～8又は7～0というように所定範

囲の 8 ビットしか取り出せないと、仮想カメラの焦点付近で最も精度良くぼけ具合を制御できるような適切な α 値を得ることができない。

例えば、Z 値のビット 19 ~ 12 を Z2 値として取り出せば、 α 値の実効的な区分け数（Z 値のしきい値数）を 256 段階にできる場合を考える。このよ
5 うな場合に、Z 値のビット 23 ~ 16 や 15 ~ 8 しか取り出せないと、 α 値の実効的な区分け数が例えば 16 段階になってしまい、画質が低下する。

これに対して、図 23 に示す手法を採用すれば、上記のように Z 値の所定範囲の 8 ビットしか取り出せないような場合にも、Z 値の任意のビット I ~ J を Z2 値として取り出すことができるようになる。従って、ニアクリップの範囲
10 や、仮想カメラの焦点の位置などに応じて、 α 値を区分けする Z 値のしきい値の設定を最適なものにすることが可能になり、より高品質な画像を生成できるようになる。

なお図 24、図 25、図 26A に、LUT1、LUT2、LUT3 の具体例を示す。また図 26B に、Z2 値を α 値に変換する LUT3 の変換特性曲線の
15 例を示す。

図 24 に示すように LUT1 は、インデックス番号として入力された Z 値のビット 15 ~ 8（ビット M ~ N）を 4 ビットだけ右にシフトする変換を行う。例えば 0x10（16 進数表現）は 0x01 に変換され、0x20 は 0x02
20 に変換される。

また図 25 に示すように LUT2 は、インデックス番号として入力された Z 値のビット 23 ~ 16（ビット K ~ L）を 4 ビットだけ左にシフトする変換を行う。例えば 0x01 は 0x10 に変換され、0x02 は 0x20 に変換され
る。

そして図 25 の Q1 に示すように、入力された Z 値が 0x0F よりも大きい
25 場合には、LUT2 の出力は 0xF0 にクランプされる。

Z 値がクランプされた場合の例を図 27 に示す。図 27 に示すように、ビット 20（ビット I ~ J 以外のビット）に 1 が立った場合には、LUT2 の出力

が $0 \times F 0$ にクランプされる。これにより $Z 2$ 値は $0 \times F 1$ になる。

例えば $LUT 2$ の出力をクランプせずに、 Z 値のビット $19 \sim 12$ をそのまま取り出してしまうと、ビット 20 に 1 が立っているにもかかわらず、 $Z 2$ 値は $0 \times 1 1$ になってしまい、被写界深度の設定が誤った設定になってしまう問題が生じる。

$LUT 2$ の出力をクランプするようにすれば、このような問題が生じるのを防止でき、被写界深度の設定を適正なものにすることができる。

しかも、このように $LUT 2$ の出力をクランプしても、図 $2 2$ から明らかなように、ニアクリップされるオブジェクト $OB 1$ やスクリーン SC の近くのオブジェクト $OB 2$ のぼけ度合いが最大値に設定されるだけであるため、画像が不自然になることもない。

2. 8 ぼかし画像の生成

さて、本実施形態では、テクスチャマッピングのバイリニアフィルタ方式（テクセル補間方式）を有効利用して、元画像（図 $1 7 A$ ）の合成対象となるぼかし画像（図 $1 7 B$ ）を生成している。

即ち、テクスチャマッピングにおいては画素の位置とテクセルの位置がずれる場合がある。

この場合に、図 $2 8$ に示すように、ポイントサンプリング方式では、画素（サンプリング点） P の色 CP （広義には画像情報）は、 P に最も距離が近いテクセル TA の色 CA になる。

一方、バイリニアフィルタ方式では、 P の色 CP は、 P の周りのテクセル TA 、 TB 、 TC 、 TD の色 CA 、 CB 、 CC 、 CD を補間した色になる。

より具体的には、 $TA \sim TD$ の座標と P の座標とに基づき、 X 軸方向の座標比 $\beta : 1 - \beta$ （ $0 \leq \beta \leq 1$ ）と、 Y 軸方向の座標比 $\gamma : 1 - \gamma$ （ $0 \leq \gamma \leq 1$ ）を求める。

この場合に、 P の色 CP （バイリニアフィルタ方式での出力色）は、下式のようなになる。

$$\begin{aligned} CP = & (1 - \beta) \times (1 - \gamma) \times CA + \beta \times (1 - \gamma) \times CB \\ & + (1 - \beta) \times \gamma \times CC + \beta \times \gamma \times CD \end{aligned} \quad (20)$$

本実施形態では、このようにバイリニアフィルタ方式では色が自動的に補間されることに着目して、ぼかし画像を生成している。

5 より具体的には図29のR1に示すように、例えばフレームバッファに描画されている元画像をテクスチャとして設定する。そして、このテクスチャ（元画像）を仮想オブジェクトにバイリニアフィルタ方式でマッピングする際に、仮想オブジェクトの頂点に与えるテクスチャ座標を、例えば（0.5、0.5）だけ右下方向にシフト（ずらす、移動）させる。このようにすることで、バイ
10 リニアフィルタ方式の補間機能により自動的に、元画像の画素の色が周囲ににじんだようなぼかし画像を生成できるようになる。

なお、画面全体をぼかす場合には、テクスチャ（元画像）をマッピングする仮想オブジェクトの形状は、画面（ぼかし領域）と同一形状に設定される。即ち、画面の頂点座標が（X、Y）＝（0、0）、（640、0）、（640、4
15 80）、（0、480）であった場合には、仮想オブジェクトの頂点座標も（X、Y）＝（0、0）、（640、0）、（640、480）、（0、480）になる。

そして、この場合に、仮想オブジェクトの頂点VX1、VX2、VX3、VX4に与えるテクスチャ座標（U、V）を、各々、（0、0）、（640、0）、（640、480）、（0、480）に設定すれば、画面の画素の位置とテクス
20 チャのテクセルの位置とがずれずに一致する。従って、画像はぼけない。

これに対して、仮想オブジェクトの頂点VX1、VX2、VX3、VX4に与えるテクスチャ座標（U、V）を、各々、（0.5、0.5）、（640.5、0.5）、（640.5、480.5）、（0.5、480.5）に設定すれば、画面の画素の位置とテクスチャのテクセルの位置とがずれるようになる。
25 従って、バイリニアフィルタ方式の補間機能により、色の補間が行われ、画像がぼけて見えるようになる。

なお、画面の一部の領域をぼかす場合には、仮想オブジェクトの形状を、そ

のぼかし領域と同一形状にすればよい。

また本実施形態では、図30のR3に示すように、元画像をテクスチャに設定し、例えば右下方向（第1のシフト方向）に0.5テクセルだけシフトしてバイリニアフィルタ方式でテクスチャマッピングを行い、第1のぼかし画像を生成する。次に、図30のR4に示すように、この第1のぼかし画像をテクスチャに設定し、例えば左上方向（第2のシフト方向）に0.5テクセルだけシフトしてバイリニアフィルタ方式でテクスチャマッピングを行い、第2のぼかし画像を生成する。或いは、以上の処理（右下方向のシフトと左上方向のシフト）を複数回繰り返す。このようにすることで、更に自然でぼかし効果の強いぼかし画像を生成できるようになる。

次に、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明する。

例えば図31Aに示すように、テクスチャ座標を0.5テクセルだけ右下方向にシフトさせて、バイリニアフィルタ方式のテクスチャマッピングを行ったとする。この場合には、上式(20)において $\beta = \gamma = 1/2$ になるため、テクセルT44、T45、T54、T55の色をC44、C45、C54、C55とすると、画素P44の色CP44は下式のようなになる。

$$CP44 = (C44 + C45 + C54 + C55) / 4 \quad (21)$$

以上から明らかなように、図31Aに示す変換により、テクセルT44の色C44（変換前の元画像の画素P44の元の色に相当）は、周りの画素P33、P34、P43、P44に対して1/4ずつしみ出すことになる。

そして、その後に図31Bに示すように、図31Aで得られた画像をテクスチャとして、テクスチャ座標を0.5テクセルだけ左上方向にシフトさせてバイリニアフィルタ方式でテクスチャマッピングを行ったとする。この場合には、図31Aの画素P33、P34、P43、P44が、図31BのテクセルT33、T34、T43、T44に対応するようになる。そして、図31AでP33、P34、P43、P44（T33、T34、T43、T44）に対して1

／4ずつしみ出した色C44が、更に1／4倍されて周りの4つの画素に対してしみ出すことになる。即ち、結局、元のT44の色C44が $1/4 \times 1/4 = 1/16$ ずつ周りにしみ出すことになる。

5 従って、図31A、図31Bの変換により、画素P33、P34、P35には、各々、色C44（フレームバッファに描かれた元画像の画素P44の元の色に相当）が1／16、2／16、1／16ずつしみ出すことになる。また、画素P43、P44、P45には、各々、色C44が2／16、4／16、2／16ずつしみ出し、画素P53、P54、P55には、各々、色C44が1／16、2／16、1／16ずつしみ出すことになる。

10 従って、図31A、図31Bの変換により、結局、図32Aに示すような平面フィルタが元画像に対して施されるようになる。この平面フィルタによれば、元画像の各画素の色がその周りに均一に広がるようになり、元画像の理想的なぼかし画像を生成できる。

15 また、図31A、図31Bの変換のセットを2回行えば、図32Bに示すような平面フィルタが元画像に対して施されるようになる。この平面フィルタによれば、図32Aよりも更に理想的なぼかし画像を生成できる。

2. 9 モニタの明るさ調整

20 さて、プレーヤが画面上のキャラクタを操作し、ダンジョンを探索するようなRPGゲームやホラーゲームでは、ダンジョンの暗い雰囲気プレーヤに感じさせるために、ゲーム画像の輝度が全体的に暗めに設定されている場合がある。

25 このような場合に、ゲーム画像が表示されるモニタの明るさ調整が、暗くなる方向に偏って設定されていると、ゲーム画像のコントラストが低下してしまい、ダンジョンの形状や模様の詳細が見えにくくなったり、ゲームの開発者が意図したものとは異なるゲーム画像が表示されたりするなどの問題が生じる。

このような問題を解決する手法として、図33に示すように、モニタ20に設けられている明るさ調整ボタン22をプレーヤが直接操作して、画面全体の

明るさを調整する手法を考えることができる。

しかしながら、この手法によると、画面の明るさを調整するためにプレーヤは、わざわざ自分の手を伸ばして調整ボタン 22 を操作しなければならず、プレーヤの利便性に欠ける。

- 5 また、このようにして、ゲームのためにモニタ 20 の明るさ調整を行ってしまうと、そのゲームを終了して他の映像ソース（テレビチューナ、ビデオ、他のゲーム）の映像を見るような場合に、明るさ調整を元に戻さなければならないなどの手間が新たに生じてしまう。

- 10 そこで、本実施形態では図 34 に示すように、プレーヤがゲームコントローラ 30 を用いて、モニタ 20 の明るさ調整（広義には表示特性の調整）のための調整データを設定できるようにする。

- 15 例えば図 34 では、プレーヤがゲームコントローラ 30 の十字キー 32 により左方向を指示すると、画面全体が明るくなるような調整データが設定され、プレーヤが十字キー 32 により右方向を指示すると、画面全体が暗くなるような調整データが設定される。

そして、このようにして設定された調整データは、プレーヤの個人データ（セーブデータ）を記憶するためのセーブ用情報記憶装置 40 にセーブされる。

- 20 そして本実施形態では、明るさ調整（表示特性の調整）により得られた調整データやセーブ用情報記憶装置 40 からロードされた調整データに基づいて、元画像の画像情報に対して変換処理が施される。

より具体的には、調整データに基づいて、ガンマ補正の変換特性（図 7 A）が特定され、その変換特性に応じたガンマ補正用 LUT（図 7 B）が作成される。そして、作成された LUT に基づいて、図 4 などでも説明した手法により元画像に対して変換処理が施される。

- 25 このようにすれば、プレーヤは、モニタ 20 に設けられている明るさ調整ボタン 22 に手を伸ばすことなく、ゲームコントローラ 30 を用いてモニタ 20 の明るさを調整できるようになる。

また、調整データはセーブ用情報記憶装置 40 に記憶され、この調整データは、プレーヤが調整処理を行ったゲームにおいてのみ有効となる。従って、そのゲームを終了して他の映像ソースの映像を見るような場合にも、明るさ調整を元に戻さなくても済むようになる。そして、再度そのゲームをプレイする場合には、セーブ用情報記憶装置 40 からロードされた調整データに基づいてモ
5 ニタ 20 の明るさ調整が行われるため、プレーヤは、明るさ調整を再度やり直さなくても済むようになる。従って、プレーヤの利便性を大幅に向上できる。

なお、セーブ用情報記憶装置 40 の記憶容量には限りがあるため、セーブする調整データのデータ量も、なるべく少ないことが望ましい。

従って、例えばガンマ補正の変換特性を特定するための調整データをセーブする場合には、ガンマ補正の変換特性を表すベジエ曲線（広義には自由曲線）の制御点のデータを、セーブ用情報記憶装置 40 にセーブすることが望ましい。即ち、例えば図 7 A の制御点 CP0、CP1、CP2、CP3 の Y 座標をセーブしたり、CP1、CP2 の Y 座標のみをセーブしたりする。このようにする
10 ことで、調整データのセーブに必要な使用記憶容量を節約でき、余った記憶容量を他の用途に使用することができるようになる。

但し、セーブ用情報記憶装置 40 の残り記憶容量に余裕がある場合には、図 7 B のガンマ補正用 LUT の内容の全てをセーブ用情報記憶装置 40 にセーブするようにしてもよい。

3. 本実施形態の処理

次に、本実施形態の処理の詳細例について、図 35 ～ 図 39 のフローチャートを用いて説明する。

図 35 は、図 12 の手法を採用した場合の本実施形態の処理例を示すフローチャートである。

まず、図 7 B に示すような変換用の LUT を作成して、VRAM に転送しておく（ステップ S1）。

次に、図 12 の D1 で説明したようにフレームバッファ上の元画像の R プレ

ーンの値をLUTのインデックス番号に設定し、LUTを用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを別バッファに描画する（ステップS2）。この時、図12のD2で説明したように、G、B値についてはマスクしておく。

5 次に、図12のD3で説明したようにフレームバッファ上の元画像のGプレーンの値をLUTのインデックス番号に設定し、LUTを用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを別バッファに描画する（ステップS3）。この時、図12のD4で説明したように、R、B値についてはマスクしておく。

10 次に、図12のD5で説明したようにフレームバッファ上の元画像のBプレーンの値をLUTのインデックス番号に設定し、LUTを用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを別バッファに描画する（ステップS4）。この時、図12のD6で説明したように、R、G値についてはマスクしておく。

15 そして最後に、別バッファに描画されている画像を、テクスチャマッピングなどを利用してフレームバッファに描画する（ステップS5）。

図36は、図13の手法を採用した場合の本実施形態の処理例を示すフローチャートである。

20 まず、図10A、図10B、図11に示すような変換用のLUTR、LUTG、LUTBを作成して、VRAMに転送しておく（ステップS10）。

次に、図13のE1で説明したようにフレームバッファ上の元画像のRプレーンの値をLUTRのインデックス番号に設定し、LUTRを用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを第1の別バッファに描画する（ステップS11）。

25 次に、図13のE3で説明したようにフレームバッファ上の元画像のGプレーンの値をLUTGのインデックス番号に設定し、LUTGを用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを第2の別バ

ツファに描画する（ステップS 1 2）。

次に、図 1 3 の E 5 で説明したようにフレームバッファ上の元画像の B プレーンの値を L U T B のインデックス番号に設定し、L U T B を用いて表示画面サイズのポリゴンにテクスチャマッピングを行い、そのポリゴンを第 3 の別バッファに描画する（ステップ S 1 3）。

次に、第 1 の別バッファに描画されている画像を、テクスチャマッピングなどを利用してフレームバッファに描画する（ステップ S 1 4）。次に、第 2 の別バッファに描画されている画像を、フレームバッファに加算描画（ α 加算）する（ステップ S 1 5）。そして最後に、第 3 の別バッファに描画されている画像を、フレームバッファに加算描画する（ステップ S 1 6）。

図 3 7、図 3 8 は、元画像の Z 値を α 値に変換し、得られた α 値に基づき元画像とぼかし画像を合成する処理（図 1 6 参照）に関するフローチャートである。

まず、元画像（透視変換後の画像）をフレームバッファに描画する（ステップ S 2 1）。この際に、Z バッファには各画素の Z 値が書き込まれることになる。

次に、Z バッファの Z 値のビット 1 5 ~ 8 を変換する L U T 1（図 2 4）、ビット 2 3 ~ 1 6 を変換する L U T 2（図 2 5）、8 ビット化された Z 値を α 値（A 値）に変換する L U T 3（図 2 6 A）を、V R A M に転送する（ステップ S 2 2）。

次に、Z 値のビット 1 5 ~ 8 を L U T 1 のインデックス番号に設定し、L U T 1 を用いて仮想ポリゴンにテクスチャマッピングを行い、その仮想ポリゴンを別バッファに描画する（ステップ S 2 3）。

次に、Z 値のビット 2 3 ~ 1 6 を L U T 2 のインデックス番号に設定し、L U T 2 を用いて仮想ポリゴンにテクスチャマッピングを行い、その仮想ポリゴンを別バッファに描画する（ステップ S 2 4）。この際、図 2 3 で説明したように、8 ビット化された Z 値の下位の 4 ビット（データ有効ビット）について

は上書きされないようにマスクしておく。

次に、ステップS 2 4で得られた8ビットのZ 2値をL U T 3のインデックス番号に設定し、L U T 3を用いて仮想ポリゴンにテクスチャマッピングを行い、その仮想ポリゴンをフレームバッファ(α プレーン)に描画する(ステップS 2 5)。

次に、ステップS 2 1でワークバッファに描画された元画像を、テクスチャ座標U、Vを(0. 5、0. 5)だけシフトしてバイリニアフィルタ方式で仮想ポリゴンにマッピングしながら、その仮想ポリゴンを別バッファに描画する(ステップS 2 6)。

次に、ステップS 2 6で別バッファに描画された画像を、テクスチャ座標U、Vを(-0. 5、-0. 5)だけシフトしてバイリニアフィルタ方式で仮想ポリゴンにマッピングしながら、その仮想ポリゴンをフレームバッファに描画する(ステップS 2 7)。この際、ステップS 2 5でフレームバッファに描画された α 値を用いて α ブレンディングを行い、元画像とぼかし画像の α 合成を行う。

以上のようにして、いわゆる被写界深度の表現が可能になる。

図3 9は、図3 4で説明した明るさ調整の処理例を示すフローチャートである。

まず、メモ리카ード(セーブ用情報記憶装置)に明るさ調整データが存在するか否かなどを判断し、明るさ調整データをロードするか否かを判断する(ステップS 3 0)。そして、ロードすると判断した場合には、メモ리카ードから明るさ調整データをロードする(ステップS 3 1)。

調整データをロードしないと判断した場合には、プレーヤが明るさ調整のオプション画面(図3 4の表示画面)を選択したか否かを判断する(ステップS 3 2)。そして、選択しなかった場合には明るさ調整データを、予め用意されている初期値に設定する(ステップS 3 3)。一方、選択した場合には、図3 4で説明したように、プレーヤからの操作データに基づいて、明るさ調整デー

タを設定（作成）する（ステップS 3 4）。そして、設定された明るさ調整データをメモリカードにセーブする（ステップS 3 5）。

次に、得られた明るさ調整データ（初期値の調整データ、設定された調整データ又はロードされた調整データ）に基づき、図4などで説明した手法により、
5 ゲーム画像を、毎フレーム、動的に変換する（ステップS 3 6）。

4. ハードウェア構成

次に、本実施形態を実現できるハードウェアの構成の一例について図40を用いて説明する。

メインプロセッサ900は、CD982（情報記憶媒体）に格納されたプログラム、通信インターフェース990を介して転送されたプログラム、或いは
10 ROM950（情報記憶媒体の1つ）に格納されたプログラムなどに基づき動作し、ゲーム処理、画像処理、音処理などの種々の処理を実行する。

コプロセッサ902は、メインプロセッサ900の処理を補助するものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算（ベクトル演算）を高速に実行する。例えば、オブジェクトを移動させたり動作（モー
15 ション）させるための物理シミュレーションに、マトリクス演算などの処理が必要な場合には、メインプロセッサ900上で動作するプログラムが、その処理をコプロセッサ902に指示（依頼）する。

ジオメトリプロセッサ904は、座標変換、透視変換、光源計算、曲面生成などのジオメトリ処理を行うものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算（ベクトル演算）を高速に実行する。例えば、座
20 標変換、透視変換、光源計算などの処理を行う場合には、メインプロセッサ900で動作するプログラムが、その処理をジオメトリプロセッサ904に指示する。

データ伸張プロセッサ906は、圧縮された画像データや音データを伸張するデコード処理を行ったり、メインプロセッサ900のデコード処理をアクセ
25 レートする処理を行う。これにより、オープニング画面、インターミッション

画面、エンディング画面、或いはゲーム画面などにおいて、MPEG方式等で圧縮された動画像を表示できるようになる。なお、デコード処理の対象となる画像データや音データは、ROM 950、CD 982に格納されたり、或いは通信インターフェース 990を介して外部から転送される。

5 描画プロセッサ 910は、ポリゴンや曲面などのプリミティブ面で構成されるオブジェクトの描画（レンダリング）処理を高速に実行するものである。オブジェクトの描画の際には、メインプロセッサ 900は、DMAコントローラ 970の機能を利用して、オブジェクトデータを描画プロセッサ 910に渡すと共に、必要であればテクスチャ記憶部 924にテクスチャを転送する。すると、描画プロセッサ 910は、これらのオブジェクトデータやテクスチャに基づいて、Zバッファなどを利用した陰面消去を行いながら、オブジェクトをフレームバッファ 922に高速に描画する。また、描画プロセッサ 910は、 α ブレンディング（半透明処理）、デプスキューイング、ミップマッピング、フォグ処理、バイリニア・フィルタリング、トライリニア・フィルタリング、アンチエイリアシング、シェーディング処理なども行うことができる。そして、1
10 フレーム分の画像がフレームバッファ 922に書き込まれると、その画像はディスプレイ 912に表示される。

 サウンドプロセッサ 930は、多チャンネルのADPCM音源などを内蔵し、BGM、効果音、音声などの高品位のゲーム音を生成する。生成されたゲーム
20 音は、スピーカ 932から出力される。

 ゲームコントローラ 942からの操作データや、メモ리카ード 944からのセーブデータ、個人データは、シリアルインターフェース 940を介してデータ転送される。

 ROM 950にはシステムプログラムなどが格納される。なお、業務用ゲームシステムの場合には、ROM 950が情報記憶媒体として機能し、ROM 9
25 50に各種プログラムが格納されることになる。なお、ROM 950の代わりにハードディスクを利用するようにしてもよい。

RAM 960 は、各種プロセッサの作業領域として用いられる。

DMAコントローラ 970 は、プロセッサ、メモリ（RAM、VRAM、ROM等）間でのDMA転送を制御するものである。

5 CDドライブ 980 は、プログラム、画像データ、或いは音データなどが格納されるCD 982（情報記憶媒体）を駆動し、これらのプログラム、データへのアクセスを可能にする。

10 通信インターフェース 990 は、ネットワークを介して外部との間でデータ転送を行うためのインターフェースである。この場合に、通信インターフェース 990 に接続されるネットワークとしては、通信回線（アナログ電話回線、ISDN）、高速シリアルバスなどを考えることができる。そして、通信回線を利用することでインターネットを介したデータ転送が可能になる。また、高速シリアルバスを利用することで、他のゲームシステムとの間でのデータ転送が可能になる。

15 なお、本発明の各手段は、その全てを、ハードウェアのみにより実現（実行）してもよいし、情報記憶媒体に格納されるプログラムや通信インターフェースを介して配信されるプログラムのみにより実現してもよい。或いは、ハードウェアとプログラムの両方により実現してもよい。

20 そして、本発明の各手段をハードウェアとプログラムの両方により実現する場合には、情報記憶媒体には、本発明の各手段をハードウェアを利用して実現するためのプログラムが格納されることになる。より具体的には、上記プログラムが、ハードウェアである各プロセッサ 902、904、906、910、930等に処理を指示すると共に、必要であればデータを渡す。そして、各プロセッサ 902、904、906、910、930等は、その指示と渡されたデータとに基づいて、本発明の各手段を実現することになる。

25 図41Aに、本実施形態を業務用ゲームシステムに適用した場合の例を示す。プレーヤは、ディスプレイ 1100上に映し出されたゲーム画像を見ながら、レバー 1102、ボタン 1104等を操作してゲームを楽しむ。内蔵されるシ

システムボード（サーキットボード）１１０６には、各種プロセッサ、各種メモリなどが実装される。そして、本発明の各手段を実現するための情報（プログラム又はデータ）は、システムボード１１０６上の情報記憶媒体であるメモリ１１０８に格納される。以下、この情報を格納情報と呼ぶ。

５ 図４１Ｂに、本実施形態を家庭用のゲームシステムに適用した場合の例を示す。プレーヤはディスプレイ１２００に映し出されたゲーム画像を見ながら、ゲームコントローラ１２０２、１２０４を操作してゲームを楽しむ。この場合、上記格納情報は、本体システムに着脱自在な情報記憶媒体であるＣＤ１２０６、或いはメモリカード１２０８、１２０９等に格納されている。

１０ 図４１Ｃに、ホスト装置１３００と、このホスト装置１３００とネットワーク１３０２（ＬＡＮのような小規模ネットワークや、インターネットのような広域ネットワーク）を介して接続される端末１３０４-１～１３０４-ｎ（ゲーム装置、携帯型電話）とを含むシステムに本実施形態を適用した場合の例を示す。この場合、上記格納情報は、例えばホスト装置１３００が制御可能な磁気ディスク装置、磁気テープ装置、メモリ等の情報記憶媒体１３０６に格納されている。

１５ 端末１３０４-１～１３０４-ｎが、スタンドアロンでゲーム画像、ゲーム音を生成できるものである場合には、ホスト装置１３００からは、ゲーム画像、ゲーム音を生成するためのゲームプログラム等が端末１３０４-１～１３０４-ｎに配送される。一方、スタンドアロンで生成できない場合には、ホスト装置

２０ １３００がゲーム画像、ゲーム音を生成し、これを端末１３０４-１～１３０４-ｎに伝送し端末において出力することになる。

なお、図４１Ｃの構成の場合に、本発明の各手段を、ホスト装置（サーバー）と端末とで分散して実現するようにしてもよい。また、本発明の各手段を実現するための上記格納情報を、ホスト装置（サーバー）の情報記憶媒体と端末の情報記憶媒体に分散して格納するようにしてもよい。

２５

またネットワークに接続する端末は、家庭用ゲームシステムであってもよいし業務用ゲームシステムであってもよい。そして、業務用ゲームシステムをネ

ットワークに接続する場合には、業務用ゲームシステムとの間で情報のやり取りが可能であると共に家庭用ゲームシステムとの間でも情報のやり取りが可能なセーブ用情報記憶装置（メモリカード、携帯型ゲーム装置）を用いることが望ましい。

- 5 なお本発明は、上記実施形態で説明したものに限らず、種々の変形実施が可能である。

例えば、本発明のうち従属請求項に係る発明においては、従属先の請求項の構成要件の一部を省略する構成とすることもできる。また、本発明の1の独立請求項に係る発明の要部を、他の独立請求項に従属させることもできる。

- 10 また、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定される画像情報としては、本実施形態で説明した情報が特に望ましいが、これに限定されるものではない。

また、本発明により実現される画像変換（ビデオフィルタ）も、図7A～図11で説明したものに限定されない。

- 15 また、調整情報をセーブ用情報記憶装置にセーブする発明においては、画像情報を変換する手法は図4で説明した手法が特に望ましいが、これに限定されない。例えば図1A、図1Bで説明した手法等により画像情報を変換してもよい。

- 20 また、奥行き値の第2の奥行き値への変換は、図23で説明したようにインデックスカラー・テクスチャマッピング用のルックアップテーブルを利用する手法で実現することが特に望ましいが、他の手法により実現してもよい。

また、奥行き値変換に使用するルックアップテーブルの変換特性も図24、図25、図26A、図6Bに示した変換特性に限定されず、種々の変形実施が可能である。

- 25 また、元画像の合成対象となるぼかし画像は、図29、図30で説明した手法により生成することが特に望ましいが、これに限定されない。例えば元画像と元画像をずらした画像を合成したり、当該フレームの元画像と前のフレーム

の元画像とを合成したりしてぼかし画像を生成してもよい。

また、テクセル補間方式を利用してぼかし画像を生成する発明も、図 2 9 ～
図 3 2 B で説明した手法に限定されない。例えば、画面全体をぼかすのではな
く、画面よりも小さいぼかし領域を設定して、その領域にある元画像をぼかす
5 ようにしてもよい。

また本実施形態では視点から近いほど奥行き値が大きくなる場合を例にとり
説明したが、視点から遠いほど奥行き値が大きくなる場合にも本発明は適用で
きる。

また、本発明は種々のゲーム（格闘ゲーム、シューティングゲーム、ロボッ
10 ト対戦ゲーム、スポーツゲーム、競争ゲーム、ロールプレイングゲーム、音楽
演奏ゲーム、ダンスゲーム等）に適用できる。

また本発明は、業務用ゲームシステム、家庭用ゲームシステム、多数のプレ
ーヤが参加する大型アトラクションシステム、シミュレータ、マルチメディア
15 端末、ゲーム画像を生成するシステムボード等の種々のゲームシステム（画像
生成システム）に適用できる。

請 求 の 範 囲

1. 画像生成を行うゲームシステムであって、

元画像の画像情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定する手段と、

- 5 元画像の画像情報がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の画像情報を変換する手段と、
を含むことを特徴とするゲームシステム。

2. 請求項 1 において、

- 10 前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とするゲームシステム。

3. 請求項 1 において、

前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とするゲームシステム。

- 15 4. 請求項 1 において、

ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2 値化、モノトーンフィルタ又はセピアフィルタの変換のための前記ルックアップテーブルを用意し、該ルックアップテーブルを用いて、元画像の画像情報に対してガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2
20 値化、モノトーンフィルタ又はセピアフィルタの変換を施すことを特徴とするゲームシステム。

5. 請求項 1 において、

- 元画像の画像情報が含む色情報の色成分を前記ルックアップテーブルのインデックス番号として設定して色情報を変換する場合において、変換後の色情報の他の色成分が描画領域に描画されないようにするためのマスク処理を行う手段を含むことを特徴とするゲームシステム。
- 25

6. 請求項 1 において、

元画像の画像情報が含む色情報の第Kの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Lの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Mの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報とを合成する手段を含むことを特徴とするゲームシステム。

7. 請求項1において、

元画像の画像情報の変換により、元画像の画像情報に応じた値の α 値が生成されることを特徴とするゲームシステム。

8. 請求項1において、

元画像の画像情報が含む奥行き値が、前記ルックアップテーブルのインデックス番号として設定されることを特徴とするゲームシステム。

9. 画像生成を行うゲームシステムであって、

元画像の各画素の奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定する手段と、

元画像の各画素の奥行き値がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定する手段と、

元画像と該元画像に対応するぼかし画像とを、各画素に設定される α 値に基づいて合成する手段と、

を含むことを特徴とするゲームシステム。

10. 請求項9において、

元画像の各画素の奥行き値を、前記奥行き値の最上位ビットよりも下位のビットI～ビットJにより構成される第2の奥行き値に変換し、前記第2の奥行き値を、インデックスカラー・テクスチャマッピング用の前記ルックアップテーブルのインデックス番号として設定することを特徴とするゲームシステム。

11. 請求項10において、

前記奥行き値のビットI～ビットJ以外のビットの値に応じて、前記第2の奥行き値を所与の値にクランプすることを特徴とするゲームシステム。

12. 請求項10において、

5 前記奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、前記ルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を前記第2の奥行き値に変換することを特徴とするゲームシステム。

10 13. 請求項10において、

前記奥行き値のビットM～ビットNを、インデックスカラー・テクスチャマッピング用の第1のルックアップテーブルのインデックス番号として設定し、前記第1のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第3の奥行き値に変換し、

15 前記奥行き値のビットK～ビットL ($K \geq I \geq L > M \geq J \geq N$) を、インデックスカラー・テクスチャマッピング用の第2のルックアップテーブルのインデックス番号として設定し、前記第2のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記
20 奥行き値を第4の奥行き値に変換し、

前記第3、第4の奥行き値に基づいて前記第2の奥行き値を求めることを特徴とするゲームシステム。

14. 請求項9において、

25 元画像をテクスチャとして設定し、該テクスチャをテクセル補間方式で仮想オブジェクトにマッピングする際に仮想オブジェクトのテクスチャ座標をシフトさせ、元画像のぼかし画像を生成することを特徴とするゲームシステム。

15. 請求項9において、

前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とするゲームシステム。

16. 請求項9において、

5 前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とするゲームシステム。

17. 家庭用ゲームのゲーム画像を生成するためのゲームシステムであって、

プレーヤがゲームコントローラを用いて入力した操作データに基づいて、モニタの表示特性を調整するための調整データを設定する手段と、

10 設定された調整データを、プレーヤの個人データを記憶するためのセーブ用情報記憶装置にセーブするセーブ手段と、

表示特性の調整により得られた調整データ又はセーブ用情報記憶装置からロードされた調整データに基づいて、元画像の画像情報に対して変換処理を施す手段と、

を含むことを特徴とするゲームシステム。

15 18. 請求項17において、

前記セーブ手段が、

画像情報の変換特性を表す自由曲線の制御点のデータを、前記調整データとしてセーブ用情報記憶装置にセーブすることを特徴とするゲームシステム。

20 19. 情報記憶媒体又は搬送波に具現化されるコンピュータ使用可能なプログラムであって、

元画像の画像情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定する手段と、

25 元画像の画像情報がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の画像情報を変換する手段と、

をコンピュータに実現させるための処理ルーチンを含むことを特徴とするプログラム。

20. 請求項19において、

前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とするプログラム。

21. 請求項19において、

5 前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とするプログラム。

22. 請求項19において、

ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換のための前記ルックアップ
10 テーブルを用意し、該ルックアップテーブルを用いて、元画像の画像情報に対してガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換を施すことを特徴とするプログラム。

23. 請求項19において、

15 元画像の画像情報が含む色情報の色成分を前記ルックアップテーブルのインデックス番号として設定して色情報を変換する場合において、変換後の色情報の他の色成分が描画領域に描画されないようにするためのマスク処理を行う手段をコンピュータに実現させるための処理ルーチンを含むことを特徴とするプログラム。

20 24. 請求項19において、

元画像の画像情報が含む色情報の第Kの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Lの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Mの色成分を前記ルックアップ
25 テーブルのインデックス番号として設定することで得られる変換後の色情報とを合成する手段をコンピュータに実現させるための処理ルーチンを含むことを特徴とするプログラム。

25. 請求項19において、

元画像の画像情報の変換により、元画像の画像情報に応じた値の α 値が生成されることを特徴とするプログラム。

26. 請求項19において、

- 5 元画像の画像情報が含む奥行き値が、前記ルックアップテーブルのインデックス番号として設定されることを特徴とするプログラム。

27. 情報記憶媒体又は搬送波に具現化されるコンピュータ使用可能なプログラムであって、

- 10 元画像の各画素の奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定する手段と、

元画像の各画素の奥行き値がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定する手段と、

- 15 元画像と該元画像に対応するぼかし画像とを、各画素に設定される α 値に基づいて合成する手段と、

をコンピュータに実現させるための処理ルーチンを含むことを特徴とするプログラム。

28. 請求項27において、

- 20 元画像の各画素の奥行き値を、前記奥行き値の最上位ビットよりも下位のビットI～ビットJにより構成される第2の奥行き値に変換し、前記第2の奥行き値を、インデックスカラー・テクスチャマッピング用の前記ルックアップテーブルのインデックス番号として設定することを特徴とするプログラム。

29. 請求項28において、

- 25 前記奥行き値のビットI～ビットJ以外のビットの値に応じて、前記第2の奥行き値を所与の値にクランプすることを特徴とするプログラム。

30. 請求項28において、

前記奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、前記ルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を前記第 2 の奥行き値に変換することを特徴とするプログラム。

3 1. 請求項 2 8 において、

前記奥行き値のビット M ～ ビット N を、インデックスカラー・テクスチャマッピング用の第 1 のルックアップテーブルのインデックス番号として設定し、前記第 1 のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第 3 の奥行き値に変換し、

前記奥行き値のビット K ～ ビット L ($K \geq I \geq L > M \geq J \geq N$) を、インデックスカラー・テクスチャマッピング用の第 2 のルックアップテーブルのインデックス番号として設定し、前記第 2 のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第 4 の奥行き値に変換し、

前記第 3、第 4 の奥行き値に基づいて前記第 2 の奥行き値を求めることを特徴とするプログラム。

3 2. 請求項 2 7 において、

元画像をテクスチャとして設定し、該テクスチャをテクセル補間方式で仮想オブジェクトにマッピングする際に仮想オブジェクトのテクスチャ座標をシフトさせ、元画像のぼかし画像を生成することを特徴とするプログラム。

3 3. 請求項 2 7 において、

前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とするプログラム。

3 4. 請求項 2 7 において、

前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴン

であることを特徴とするプログラム。

35. 情報記憶媒体又は搬送波に具現化されるコンピュータ使用可能なプログラムであり、家庭用ゲームのゲーム画像を生成するためのプログラムであって、プレーヤがゲームコントローラを用いて入力した操作データに基づいて、モニタの表示特性を調整するための調整データを設定する手段と、

設定された調整データを、プレーヤの個人データを記憶するためのセーブ用情報記憶装置にセーブするセーブ手段と、

表示特性の調整により得られた調整データ又はセーブ用情報記憶装置からロードされた調整データに基づいて、元画像の画像情報に対して変換処理を施す手段と、

をコンピュータに実現させるための処理ルーチンを含むことを特徴とするプログラム。

36. 請求項35において、

前記セーブ手段が、

画像情報の変換特性を表す自由曲線の制御点のデータを、前記調整データとしてセーブ用情報記憶装置にセーブすることを特徴とするプログラム。

37. 画像生成を行うための画像生成方法であって、

元画像の画像情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、

元画像の画像情報がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の画像情報を変換することを特徴とする画像生成方法。

38. 請求項37において、

前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とする画像生成方法。

39. 請求項37において、

前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とする画像生成方法。

40. 請求項37において、

ガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換のための前記ルックアップテーブルを用意し、該ルックアップテーブルを用いて、元画像の画像情報に対してガンマ補正、ネガポジ反転、ポスタリゼーション、ソラリゼーション、2値化、モノトーンフィルタ又はセピアフィルタの変換を施すことを特徴とする画像生成方法。

10 41. 請求項37において、

元画像の画像情報が含む色情報の色成分を前記ルックアップテーブルのインデックス番号として設定して色情報を変換する場合において、変換後の色情報の他の色成分が描画領域に描画されないようにするためのマスク処理を行うことを特徴とする画像生成方法。

15 42. 請求項37において、

元画像の画像情報が含む色情報の第Kの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Lの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報と、色情報の第Mの色成分を前記ルックアップテーブルのインデックス番号として設定することで得られる変換後の色情報とを合成することを特徴とする画像生成方法。

43. 請求項37において、

元画像の画像情報の変換により、元画像の画像情報に応じた値の α 値が生成されることを特徴とする画像生成方法。

25 44. 請求項37において、

元画像の画像情報が含む奥行き値が、前記ルックアップテーブルのインデックス番号として設定されることを特徴とする画像生成方法。

4 5. 画像生成を行う画像生成方法であって、

元画像の各画素の奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、

5 元画像の各画素の奥行き値がインデックス番号として設定された前記ルックアップテーブルを用いて、仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、元画像の各画素の奥行き値に応じた値に各画素の α 値を設定し、

元画像と該元画像に対応するぼかし画像とを、各画素に設定される α 値に基づいて合成することを特徴とする画像生成方法。

10 4 6. 請求項 4 5 において、

元画像の各画素の奥行き値を、前記奥行き値の最上位ビットよりも下位のビット I ~ ビット J により構成される第 2 の奥行き値に変換し、前記第 2 の奥行き値を、インデックスカラー・テクスチャマッピング用の前記ルックアップテーブルのインデックス番号として設定することを特徴とする画像生成方法。

15 4 7. 請求項 4 6 において、

前記奥行き値のビット I ~ ビット J 以外のビットの値に応じて、前記第 2 の奥行き値を所与の値にクランプすることを特徴とする画像生成方法。

4 8. 請求項 4 6 において、

20 前記奥行き値を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、前記ルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を前記第 2 の奥行き値に変換することを特徴とする画像生成方法。

4 9. 請求項 4 6 において、

25 前記奥行き値のビット M ~ ビット N を、インデックスカラー・テクスチャマッピング用の第 1 のルックアップテーブルのインデックス番号として設定し、前記第 1 のルックアップテーブルを用いて仮想オブジェクトに対してインデッ

クスカラー・テクスチャマッピングを行い、前記奥行き値を第3の奥行き値に変換し、

5 前記奥行き値のビットK～ビットL ($K \geq I \geq L > M \geq J \geq N$) を、インデックスカラー・テクスチャマッピング用の第2のルックアップテーブルのインデックス番号として設定し、前記第2のルックアップテーブルを用いて仮想オブジェクトに対してインデックスカラー・テクスチャマッピングを行い、前記奥行き値を第4の奥行き値に変換し、

前記第3、第4の奥行き値に基づいて前記第2の奥行き値を求めることを特徴とする画像生成方法。

10 50. 請求項45において、

元画像をテクスチャとして設定し、該テクスチャをテクセル補間方式で仮想オブジェクトにマッピングする際に仮想オブジェクトのテクスチャ座標をシフトさせ、元画像のぼかし画像を生成することを特徴とする画像生成方法。

51. 請求項45において、

15 前記仮想オブジェクトが、表示画面サイズのポリゴンであることを特徴とする画像生成方法。

52. 請求項45において、

前記仮想オブジェクトが、表示画面を分割したブロックのサイズのポリゴンであることを特徴とする画像生成方法。

20 53. 家庭用ゲームのゲーム画像を生成するための画像生成方法であって、

プレーヤがゲームコントローラを用いて入力した操作データに基づいて、モニタの表示特性を調整するための調整データを設定し、

設定された調整データを、プレーヤの個人データを記憶するためのセーブ用情報記憶装置にセーブし、

25 表示特性の調整により得られた調整データ又はセーブ用情報記憶装置からロードされた調整データに基づいて、元画像の画像情報に対して変換処理を施すことを特徴とする画像生成方法。

54. 請求項53において、

画像情報の変換特性を表す自由曲線の制御点のデータを、前記調整データとしてセーブ用情報記憶装置にセーブすることを特徴とする画像生成方法。